P/1877/98

Kryptograficzna ochrona poczty komputerowej

miesięcznik profesjonalistów

3
1998

informatyka

Cena: 6,0 zł. ISSN 0542 9951 WYDAWNICTWO SIGMA NOT

Pismo informatyczne - ukazuje się od 1965 roku

Inżynieria wymagań, czyli jak unikać zbędnej pracy

Zasady tworzenia systemów informatycznych

Zagrożenia dla zamierzeń informatycznych

SYSTEMY TRWAŁEJ ARCHIWIZACJI DOKUMENTÓW I DANYCH

Oferujemy ekonomiczne, sprawne w obsłudze systemy archiwizacji, zarządzania i wyszukiwania informacji tworzące rodzinę rozwiązań

mis ARCHI+

- Tworzenie efektywnego i łatwego w obsłudze archiwum na trwałych nośnikach CD-R, co zapewnia absolutne bezpieczeństwo danych. Jest to szczególnie ważne w przypadku danych finansowo-księgowych. Zapis na nośnikach CDR jest zgodny z Ustawą o Rachunkowości z 1994 r.
- Wysoka efektywność wyszukiwania dokumentów w obrębie dużego archiwum, składającego się z kilkuset płytek oraz natychmiastowy dostęp do całego archiwum wszystkim użytkownikom sieci, niezależny od fizycznego położenia poszczególnych nośników.
- Dane zgromadzone w archiwum są skompresowane, dzięki czemu na 1 nośniku CD-R można zgromadzić aż 3 GB raportów ASCII.
- Możliwość elastycznego opisu archiwizowanych dokumentów za pomocą słów kluczowych, wartości, dat i innych.
- Możliwość przeszukiwania pełnotekstowego, według słów kluczowych, tekstów pochodzących z funkcji OCR (czas wyszukiwania dowolnego dokumentu nie przekracza 3 sekund.
- Możliwość odczytu dowolnego nośnika wchodzącego w skład archiwum na dowolnym komputerze z systemem Windows.
- Aplikacja przeszukująca i odtwarzająca zawiera wbudowane przeglądarki dla formatów: Word, WordPerfect, 1-2-3, Excel, Quattro, PCX, TIFF, G3 i G4, DCX, GIF, BMP, Targa, EPS, WPG, PICT, JPEG, CALS raster, Sun raster, CLP, ICO, GEM, MSP, CAD, Kofax, GX2, MacPaint, RLE, Photoshop, LaseData, XBM, XPM, Brookout, HALO, IFF, IOC, XWD i inne.
- Systemy stworzone w oparciu o oprogramowanie firmy IMR stosowane są przez ponad 300.000 klientów na całym świecie (np. Boeing, US Navy, Black&Decker).

System składa się z następujących elementów:

- Stanowisko skanowania, tworzenia bazy, wyszukiwania i odtwarzania dokumentów:
  - skaner dokumentów (np. Fujitsu),
  - nagrywarka płyt CD-R (np. Yamaha),
  - system operacyjny wraz z wybranymi modułami oprogramowania firmy IMR.
- Stanowisko udostępniania archiwum dokumentów w sieci:
  - CD-ROM Serwer lub zmieniacz płyt CD (np. jukebox NSM)
  - system operacyjny wraz z oprogramowaniem zarządzającym nośnikami CD-R oraz udostępniającym archiwum w sieci (np. WinNT + IXOS)

ARCHI+
ZAKŁADY ENERGETYCZNE
Kompleksowy system archiwizacji raportów, rozliczeń z klientami, umów, zleceń, projektów, baz danych itp. dostosowany do specyfiki zakładów energetycznych

ARCHI+
FINANSOWO-KSIĘGOWY
Kompleksowy system tworzenia trwałego archiwum dokumentów finansowo-księgowych (zapis zgodny z Ustawą o Rachunkowości z 1994 r.)

ARCHI+
BIURA PROJEKTOWE
Kompleksowy system archiwizacji projektów CAD, dokumentacji, katalogów producentów, wykonawców, materiałów itp. w biurach projektowych

BĄDŹ PANEM SWEGO CZASU

mis
Management Information Systems Sp. z o.o

UWAGA! Nowy adres siedziby naszej firmy

MIS Sp. z o.o., 51-502 Wrocław, ul. Mydlana 1, tel./fax (0-71) 3450110, 3450111

informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124

P1877/98

POLITECHNIKA ŚLĄSKA Biblioteka Główna

nr 3 1998
Marzec
Rok wydania XXXIV

REDAGUJE ZESPÓŁ:
dr Lesław **WAWRZONEK**
(redaktor naczelny)
Alina **KLEPACZ**
(sekretarz redakcji)
redaktorzy:
mgr Zdzisław **ŻURAKOWSKI**
dr Ewa **ŁUKASIK**
współpraca:
Rafał **MAŚLANA**
Ewa **DULNA** (korekta)
Joanna **OTTO** (adm. redakcji)

KOLEGIUM REDAKCYJNE:
prof.dr hab. Leonard BOLC
mgr inż. Piotr FUGLEWICZ
prof. dr hab. Jan GOLIŃSKI
dr inż. Zenon KULPA
prof. dr inż. Jan MULAWKA
prof. dr hab. Wojciech OLEJNICZAK
mgr. inż. Jan RYŻKO
dr Witold STANISZKIS
dr inż. Jacek STOCHLAK
prof. dr hab. Maciej STOLARSKI
prof. dr hab. Zdzisław SZYJEWSKI
prof. dr hab. inż.
Ryszard TADEUSIEWICZ
prof. dr hab. Jan WĘGLARZ
PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab. Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

REDAKCJA:
00-950 Warszawa,
ul. Ratuszowa 11, p. 644
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail: informat@pol.pl
www.pol.pl/informatyka/

PRENUMERATA:
tel. 40-35-89, 40-30-86
Prenumeratę przyjmujemy również
w sieci INTERNET:
WWW.pol.pl./sigma_not
E-mail: kolpor.sigma@pol.pl

Materiałów nie zamówionych redakcja nie zwraca.

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach 3½" lub pocztą elektroniczną - w edytorach Word.
Po szczegółowe informacje dla Autorów prosimy zwracać się do redakcji.

Redakcja nie ingeruje w treść i formę ogłoszeń oraz innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

Ogłoszenia przyjmują:
- Redakcja, tel. 619-11-61
- Dział Reklamy i Marketingu
00-950 Warszawa, ul. Mazowiecka 12
tel.: 827-43-66, fax: 826-80-16

Okładka:
AGAT, Jerzy Burski i Andrzej Jacyszyn,
Łamanie:
Alina Klepacz, program PageMaker

Druk:
Drukarnia SIGMA NOT Sp. z o.o.

W numerze:

INFORMACJE 3

SYSTEMY INFORMATYCZNE

Przetwarzanie centralne czy rozproszone? cz. 2 6
Piotr Kowalski

Inżynieria wymagań, czyli jak unikać zbędnej pracy 14
Janusz Górski

Zasady tworzenia systemów informatycznych 20
Grażyna Hołodnik-Janczura

PUBLIKACJE

Obiektowość w bazach danych – koncepcje, nadzieje i fakty (3) 26
Kazimierz Subieta

KONFERENCJE

Zagrożenia dla zamierzeń informatycznych 34
Marian Kuraś, Agnieszka Zając

Kryptograficzna ochrona poczty komputerowej 38
Piotr Szpryngier

Zależności temporalne w modelowaniu i analizie procesów gospodarczych 42
Krzysztof Kania, Stanisław Kędzierski, Jerzy Gołuchowski

BIULETYN PTI 46

W najbliższych numerach ukażą się artykuły:

Komputer w roli nauczyciela języków obcych – Grażyna Hołodnik-Janczura

Przetwarzanie centralne czy rozproszone? cz. 3: stawką - wiarygodna i pełna informacja – Piotr Kowalski

Tajemnica statystyczna – Georges Als

Obliczenia molekularne - nowy kierunek technik informacyjnych – Jan Mulawka, Piotr Wąsiewicz

Niefortunny przykład propagandy sukcesu

Nasze władze samodzielnie oraz za pomocą niezależnej prasy, usiłując dodać sobie otuchy, powołują się ostatnio dość często na „pogrążoną w kryzysie gospodarczym" Republikę Czeską. „Coraz rzadziej mówi się dziś o czeskim cudzie gospodarczym, częściej zaś o symptomach kryzysu" - pisze w stylu prawie zapomnianej już nowomowy korespondent Prawa i Gospodarki z Bratysławy (dlaczego nie z Pragi?).

Porównanie wskaźników makroekonomicznych nie jest jednak dla nas pocieszające. W Czechach w 1997 roku bezrobocie i inflacja były dwa razy mniejsze niż Polsce. W przeliczeniu na jednego mieszkańca mamy o 40% niższy dochód narodowy oraz dwa i pół raza mniejsze nakłady na naukę, a w rezultacie czescy naukowcy opublikowali w 1995 r. ponad dwa razy więcej prac. Dodajmy, że przy liczbie mieszkańców 38,6 mln w Polsce i 10,3 mln w Czechach wartości eksportu obu krajów były w 1997 r. prawie równe.

Czytelników „Informatyki" pewnie bardziej interesuje porównanie Polski z południowym sąsiadem w dziedzinie technologii informatycznych. Niektóre różnice ilustrują aż nadto wymownie liczby i fakty, a inne - subiektywne wrażenia.

W raporcie „Polityki" na temat polskiej komputeryzacji największe zdziwienie czytelników, z którymi rozmawiałem, wzbudzały dane o nasyceniu informatyką Czech i Polski. Opisane tam malwersacje i nieudolność naszej administracji w ciągu ostatnich ośmiu lat nie szokowały, być może dlatego, że zaczynamy traktować je jako stały element naszego krajobrazu.

Mogłoby się wydawać, że skoro liczba ludności w Polsce jest czterokrotnie większa niż w Czechach, to kupujmy znacznie więcej komputerów różnego typu. Według raportu European Information Technology Observatory dużych i średnich systemów jest w Czechach znacznie więcej niż w Polsce, a PC-ów kupujemy tyle samo. Według tego raportu globalne wydatki na technologie informatyczne są w Polsce mniejsze.

Liczne przedstawicielstwa światowych firm komputerowych były otwierane w Czechach znacznie wcześniej niż u nas. Na przykład, kiedy przedstawiciele Autodesku w 1995 szukali pomieszczenia na swoje biuro Warszawie, to w Czechach istniał już od pięciu lat oddział tej firmy, wydający własne fachowe pismo. Także obrotów oddziałów innych koncernów w Czechach mogą pozazdrościć polscy koledzy. Trudno się dziwić skoro chłonność czeskiej gospodarki na produkty i usługi informatyczne jest kilkakrotnie większa.

Nie inaczej jest w branży wydawniczej. Zachodni wydawcy prasy komputerowej, mający swoje czasopisma w obu krajach, nie mogą się nadziwić, że wielkości nakładów tych samych tytułów są zbliżone. Wynika stąd, że namiestnicy tych wydawców w Polsce są nieudolni albo pracują w kraju bardziej zacofanym cywilizacyjnie. W Czechach pięciotysięczny nakład popularnej książki komputerowej uważa się za standardowy, zaś u nas - 1500.

Wiele też mówi porównanie krajowych systemów komputerowych do obsługi podatków. W Czechach został on wykonany przez dużą amerykańską firmę w ciągu półtora roku i działa już od 1994 r. Ta sama firma odpadła u nas w przetargu na wykonanie systemu Poltax. W niewyjaśnionych okolicznościach wygrał, będący w kłopotach finansowych, francuski producent komputerów, Bull. W rezultacie nasz kraj stracił prawie 100 mln dolarów na budowę systemu, a około 15 mld USD przepadło w ciągu pięciu lat w wyniku braku kontroli ściągania podatków. Po ośmiu latach od podpisania kontraktu mamy komputerowy system podatkowy, który w klasyfikacji zabytków byłby określony mianem „trwałej ruiny". Odpowiedzialny za budowę tej kosztownej ruiny (wg raportu NIK jest nim obecny wicepremier L. Balcerowicz) przestrzega nas przed czeską „katastrofą" gospodarczą.

Porównanie metod komputeryzacji służby zdrowia także nie nastraja optymistycznie. Czesi stworzyli program do obsługi szpitali, który sukcesywnie w nich wdrażają. My natomiast, pomijając przewlekłą chorobę, jaką jest realizacja Rejestru Usług Medycznych, „skomputeryzowaliśmy" kilkadziesiąt szpitali za kredyt Banku Światowego. Kupiono za 30 mln USD bardzo drogie, tzw. markowe komputery PC z oprogramowaniem Office. Oczywiście, że bez żadnych sieci, nie mówiąc już o specjalistycznym oprogramowaniu. Taką „komputeryzację" może zrobić każda firma w Polsce, bez angażowania największego koncernu z tej branży na świecie.

Pewną miarą siły i ekspansji branży komputerowej kraju jest jego reprezentacja na hanowerskich targach CeBIT. W tym roku polskie firmy wykupiły tam skromne 200 m^2, natomiast Czechy, kraj czterokrotnie mniej liczny i przeżywający ponoć trudności gospodarcze - 365 m^2. Pozostając przy targach, gdyby ktoś chciał obejrzeć z bliska czeski „kryzys", może wybrać się jesienią na targi INVEX w Brnie. Są one pod względem powierzchni i liczby zwiedzających większe niż nasz Infosystem, KomputerExpo i Softarg razem.

Kto dłużej żyje, ten wie, że choć władza się zmienia, to dworacy są ci sami. Wmawiali już nam, że jesteśmy 10. potęgą przemysłową świata, drugą Japonią, a ostatnio, przodownikiem przemian w Europie Środkowo-Wschodniej. Jeżeli nasycenie produktami i usługami informatycznymi jest wyznacznikiem rozwoju i siły gospodarki, to przykład Republiki Czeskiej, mający pokrzepiać nasze serca, nie został wybrany zbyt fortunnie.

Leszek Jaworowski

JDEdward w Kostrzyn Paper

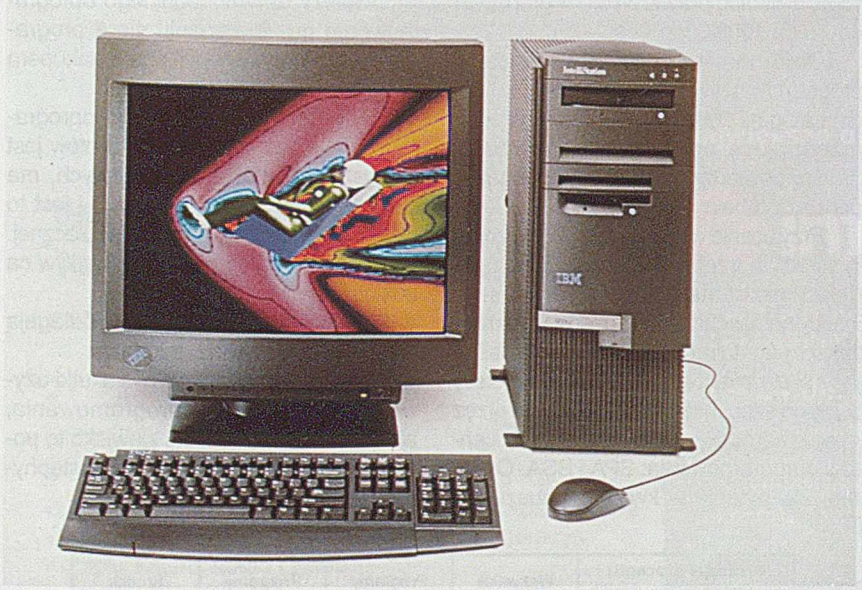
Warszawska firma HOGART wygrała przetarg na wdrożenie zintegrowanego systemu JDEdwards w Kostrzyn Paper. Zgodnie z harmonogramem ogólnym podstawowe elementy księgowe, umożliwiające raportowanie fiskalne oraz podstawowe elementy obsługi sprzedaży z elementami gospodarki magazynowej, będą wdrażane od końca marca 1998 r. W kolejnym etapie zostanie zaadaptowana księgowość zarządcza, sprzedaż, gospodarka materiałowa, zakupy i produkcja. Następnym elementem będzie wdrożenie funkcji niezbędnych do utrzymania ruchu. Po uruchomieniu systemu nastąpi czteromiesięczny okres próbny, podczas którego będą zbierane opinie użytkowników, aby później ostatecznie zaadaptować system do potrzeb Kostrzyn Paper.

Lepszy dostęp do baz danych za pomocą WWW

Novell i Oracle nawiązały strategiczną współpracę, mającą na celu oferowanie lepszych możliwości w zakresie dostępu do baz danych oraz tworzenia na platformie NetWare aplikacji wykorzystujących zalety WWW. Opracowana przez Oracle architektura Network Computing Architecture umożliwiła tworzenie aplikacji współpracujących z WWW poprzez zastosowanie standardowych wywołań HTTP w serwerach WWW. Oracle Web Application Server obsługuje zdalne wywołanie aplikacji serwerowych zwanych zasobnikami (cartridges), które mogą być napisane w Javie. Tego typu zasobniki mogą służyć do wyszukiwania informacji w bazie danych oraz wykonywania całego szeregu zadań mających na celu dostarczanie użytkownikowi informacji za pośrednictwem przeglądarek WWW. Architektura OSA (open solution architecture) przyczynia się do rozszerzenia środowiska tworzenia aplikacji internetowych Oracle, udostępniając interfejsy Javy, które zwiększają poziom zabezpieczenia zasobników i ułatwiają administrowanie nimi. Zasobniki Javy można tworzyć przy użyciu zestawu OSA SDK dla programistów, korzystającego z bogatych usług sieciowych.

Nowa stacja robocza IBM z serii IntelliStation NT

Stacja IntelliStation M Pro wykorzystuje procesory Pentium II (233, 260 lub 300 Mhz) pamięci SDRAM z korekcją błędów o maksymalnej pojemności 512 MB oraz twarde dyski o pojemności do 25 GB. Ma możliwość przetwarzania dwuprocesorowego. Zastosowanie chipsetu Intela: 44 LX pozwala na użycie synchronicznej pamięci DRAM, portu graficznego AGP oraz szybkich akceleratorów graficznych. Po raz pierwszy w tej klasie stacji zastosowano szybki dwukanałowy sterownik SCSI oparty na układzie Adaptec 7895, gwarantującym szybki transfer i duże bezpieczeństwo danych. Stację można wyposażyć w twarde dyski IBM - Wide Ultra SCSI S.M.A.R.T między innymi: 4.2 GB (5400 rpm), 4.5 GB (7200 rpm) lub 9.1 GB (7200 rpm). Najbardziej zaawansowany model stacji Intellistation M Pro posiada kartę graficzną Intergraph Intense 3D Pro2200/4T z 20 MB pamięci. W teście CDRS-03 uzyskał on wynik 41.4 ramek na sekundę.



IBM IntelliStation

IBM propaguje handel elektroniczny

IBM zaprezentował specjalną ofertę obejmującą serwery z rodziny IBM RS/6000 wraz z systemem UNIX-owym AIX 4.3. System ten oferuje wszystkie kluczowe funkcje potrzebne do prowadzenia interesów w sieci elektronicznej. System AIX 4.3 umożliwia administrowanie systemem za pomocą przeglądarki WWW, ochronę transakcji dokonywanych poprzez Internet, obsługę wirtualnych - prywatnych sieci zgodnie z obowiązującymi standardami, szerszą współpracę z komputerami sieciowymi (NC). Posiada kompilator JIT (just in time) języka Java zapewniający wydajność zbliżoną do szybkości działania programów opartych na natywnym kodzie C.

Zintegrowany system routerowo-modemowy do sieci kablowych

Cisco Systems zaprezentował serię produktów do przesyłania danych w sieciach telewizji kablowej. Pierwszymi urządzeniami z tej serii są uniwersalny router szerokopasmowy Cisco uBR 7246 oraz karty modemowe MCNS (Multimedia Cable Network System). Cisco uBR 7246 jest pierwszym zintegrowanym systemem routerowo-modemowym zgodnym z MCNS. Umożliwia on operatorom sieci kablowych zwiększenie bezpieczeństwa, wydajniejsze wykorzystanie pasma oraz centralne administrowanie i sterowanie infrastrukturą. Pozwala to na zaoferowanie takich usług, jak szybki dostęp do Internetu, strumieniowe przesyłanie wideo, bezpieczna transmisja danych oraz obsługa wirtualnych sieci prywatnych.

Bazujący na routerze Cisco 7200 zestaw Cisco uBR 7246 zapewnia przepustowość przekraczającą 600 Mb/sek. i szybkość przełączania 100 tys. pakietów na sekundę. Cisco oferuje również pełen pakiet oprogramowania Cisco IOS, ułatwiający administrowanie zasobami sieci, przydzielanie poszczególnym aplikacjom zróżnicowanych poziomów usług oraz realizację dodatkowych funkcji, takich jak przełączanie NetFlow czy

obsługa protokołu HSRP (*Hot Standby Router Protocol)* w celu uzyskania ciągłego połączenia o gwarantowanej jakości. Urządzenie to można konfigurować i nadzorować za pośrednictwem sieci WWW.

Najlepszy program antywirusowy roku 1997

PC Magazine przyznało programowi Norton AntiVirus 2.0 do Windows 95 nagrodę „Najlepszy program roku 1997" w kategorii programów antywirusowych. Wprowadzając w 1996 roku program Norton AntiVirus 2.0, a później Norton AntiVirus 4.0, Symantec udostępnia wiele innowacji, skupionych na rozszerzonej funkcjonalności Internetu, prostocie obsługi, spójnej ochronie na bieżąco oraz bezpieczeństwie. Do najważniejszych opcji należą: LiveUpdate, która zapewnia aktualizacje definicji wirusów po naciśnięciu jednego przycisku, Striker, najnowszej generacji mechanizm wykrywania wirusów, który znajduje również wyrafinowane, zaszyfrowane wirusy polimorficzne oraz kreator Repair Wizard, który przeprowadza użytkowników przez proces usuwania wirusów.

Platinum SQL i Platinum for Windows po polsku

Platinum Software Corporation, dostawca systemów finansowych opartych na technologii klient/serwer, ogłasza wprowadzenie na polski rynek swoich dwóch programów, które będą sprzedawane i wdrażane przez Qumak International.

Obecnie zakończono lokalizację systemu Platinum for Windows i firma Qumak International jest na etapie jego wdrażania w kilku przedsiębiorstwach w Polsce. Lokalizacja Platinum SQL jest obecnie w toku i zostanie zakończona w pierwszym kwartale 1998 roku.

Zabezpieczenie biura i danych

Radiotechnika Marketing Sp. z o.o. oferuje zintegrowany system zabezpieczenia biura i danych komputerowych „VizyOffice". Jest to pierwsze w Polsce rozwiązanie służące do zabezpieczenia i kontroli dostępu do pomieszczeń oraz kontroli dostępu do komputerów. System ten kontroluje dostęp do baz danych, ważnych stanowisk komputerowych i pomieszczeń specjalnego znaczenia, przy czym prawo wstępu do ośrodka obliczeniowego czy laboratorium nie uprawnia jeszcze do pracy na komputerach tam się znajdujących. Prawa wstępu do pomieszczeń oraz pracy na określonych komputerach są zapisane przez administratora na kluczu pracownika -plastikowej karcie z chipem. System posiada również funkcje nocnej kontroli obchodu strażników oraz kontroli czasu pracy.

Wyniki sondażu „Prawo autorskie na polskim rynku oprogramowania"

Stowarzyszenie Polski Rynek Oprogramowania PRO przeprowadziło w 97 r. sondaż, mający na celu zbadanie stosowania prawa autorskiego na polskim rynku oprogramowania i ocenę jego funkcjonowania.

Można ogólnie stwierdzić, że respondenci tego sondażu stanowili stosunkowo wszechstronną reprezentację użytkowników oprogramowania.

Jedno z pytań sondażu miało przedstawić zdanie użytkowników oprogramowania na temat skali piractwa w Polsce w trzech kategoriach:

1. wszystkie programy (łącznie),
2. programy polskie,
3. programy importowane (w tym ich spolszczone wersje).

Na pytanie to udzielono 727 odpowiedzi na 837 uczestników sondażu (86,9%).

Łącznie w ocenie ponad 46% z 727 respondentów wszystkie programy są używane nielegalnie w granicach 50 - 75%, a według prawie 31% - w granicach od 75 do 100%.

Obliczono średnią wartość procentu programów używanych nielegalnie w każdej kategorii (za pomocą metody środka przedziału, z pominięciem odpowiedzi „trudno powiedzieć"). Wyniki przedstawia poniższa tabela:

Przy ocenie wyników pytania o rozmiary piractwa trzeba pamiętać, że respondenci przedstawiali swoje własne, intuicyjne szacunki. Warto jednak zauważyć, że uśredniony wynik (64,0%) określający skalę piractwa w Polsce jest niższy o 7% od szacunków publikowanych przez międzynarodowe organizacje producentów oprogramowania: SPA i BSA. Ostatnie, 22. pytanie w kwestionariuszu dla użytkowników oprogramowania zawierało 23 zdania, które respondent miał ocenić w skali:

A. zgadzam się całkowicie;
B. zgadzam się, ale nie do końca;
C. nie zgadzam się wcale.

Respondenci najczęściej zgadzali się ze zdaniami (w nawiasie podano procent odpowiedzi „zgadzam się całkowicie"):

– Nabywca powinien mieć możliwość zapoznania się z programem przed podjęciem decyzji o zakupie (93,9%);
– Nabywca licencji powinien mieć możliwość zapoznania się z zasadami licencji przed dokonaniem ostatecznego zakupu (90,9%);
– Nielegalne kopiowanie oprogramowania jest kradzieżą (76,8%).

Największy sprzeciw budziły zdania (w nawiasie podano procent odpowiedzi „nie zgadzam się wcale"):

– Nielegalne kopiowanie programów jest czynem o znikomym stopniu szkodliwości społecznej (51,2%);
– Sprzedaż podręczników obsługi popularnych programów ma wpływ na piracki rynek. Bez nich programy byłyby bezużyteczne, a legalni posiadacze używają i tak podręczników użytkownika dołączonych do programu (46,2%);
– Strona WWW jest programem komputerowym (44,2%).

Za pomocą analizy czynnikowej ustalono pięć istotnych czynników dla większości respondentów. Są one wymienione poniżej w kolejności ich „wagi" - poziomu wyjaśnienia 23 zdań w pytaniu 22 ankiety:

- Należy używać legalnego oprogramowania po zapoznaniu się z programem i zasadami licencji przed zakupem programu.
- Stosowanie nielegalnego oprogramowania przez uczniów i studentów jest korzystne dla firm software'owych, ma istotne znaczenie marketingowe i jest to czyn o znikomej szkodliwości społecznej.
- Cena oprogramowania ma wpływ na poziom piractwa.
- Internet i strony WWW podlegają ochronie przez prawo autorskie.
- Istnieją różne przyczyny i skutki używania nielegalnego oprogramowania; ale w każdym przypadku zjawisko to powinno być ścigane wszelkimi dostępnymi prawnie środkami.

Przedział procentu programów używanych nielegalnie	Wszystkie programy	Programy polskie	Programy importowane	Środek przedziału
0 - 25%	43	101	49	12,5
25 - 50%	96	237	101	37,5
50 - 75%	337	232	252	62,5
75 - 100%	223	103	274	87,5
Liczba odpowiedzi	699	673	676	
Średnia wartość procentu programów używanych nielegalnie	64,0%	50,0%	65,3%	

Przedstawione powyżej wyniki dotyczą jedynie wybranych pytań z sondażu skierowanego do użytkowników oprogramowania. Obszerne dane o wynikach całego badania zawierać będzie zbiorczy raport analityczny.

System Informixa za darmo

Od 1 stycznia, przez trzy miesiące, Sun Microsystems i Informix Software oferują wspólne rozwiązanie systemu przeznaczonego do obsługi wydziałów w przedsiębiorstwach, grup roboczych oraz małej i średniej wielkości firm. W okresie promocji zestaw składający się z serwera Sun E450 i systemu zarządzania bazą danych INFORMIX-Dynamic Server, Workgroup Edition kosztuje taniej niż serwer E450 kupowany osobno.

Lokalna, cyfrowa transmisja bezprzewodowa

Spółka **Bonair S.A.** wprowadza na polski rynek rodzinę cyfrowych urządzeń **AirLink** i **AirLink PRO.** Umożliwą one bezprze-wodową transmisję głosu i danych dla tych rejonów Polski, w których operatorzy publicznych sieci telekomunikacyjnych nie mogą zaoferować swoich usług.

Instalując urządzenia AirLink, można zapewnić łączność na przykład między zlokalizowanym w centrum miasta biurem i wybudowanym poza miastem magazynem. Urządzenia te otrzymały już homologację Ministra Łączności. W pierwszym dniu targów Komputer Expo firma Bonair, integrator i dostawca rozwiązań z zakresu zaawansowanej technologii otrzymała za urządzenia AirLink nagrodę Grand Prix w kategorii urządzeń peryferyjnych i osprzętu.

Komputeryzacja w krakowskiej POLFIE

Przedsiębiorstwo FORTECH z Krakowa i Krakowskie Zakłady Farmaceutyczne "POLFA" S.A. podpisały umowę o dostawie i instalacji komputera firmy Siemens Nixdorf RM600 E60 w "POLFIE". Komputer RM600 E60 jest serwerem typu High End, który oferuje możliwości typu "Clustering" oraz "High availibility".

KZF "POLFA" S.A. wybrały sprzęt firmy Siemens Nixdorf w przetargu, m.in. ze względu na referencje międzynarodowych koncernów farmaceutyczno - chemicznych.

Serwer RM600 E60 będzie pracował na aplikacjach firmy FORTECH pod bazą danych Informix On Line Dynamic Server 7.21 i będzie obsługiwał Zakładowy System Informatyczny do zarządzania całością przedsiębiorstwa. Jest to pierwszy tego typu serwer w Polsce.

Największy europejski projekt zarządzania kadrami

Deutsche Telekom będzie wdrażać system zarządzania kadrami firmy SAP. Moduł zarządzania kadrami realizowany w Deutsche Telekom będzie obejmował 10 000 użytkowników oraz 200 000 rekordów danych podstawowych. Oznacza to, że takie funkcje R/3, jak administrowanie kadrami, zarządzanie czasem pracy, rekrutacja czy rozliczanie podróży służbowych będą współpracować razem z systemem informacyjnym planowania kadr.

Doświadczenia uzyskane podczas realizacji tego projektu zostaną wykorzystane w przyszłych systemach standardowych SAP.

Przetwarzanie centralne czy rozproszone?

Część II: Cel – elastyczność i efektywność

Piotr Kowalski

Decydując się na wybór konkretnej technologii przetwarzania danych, zdajemy sobie sprawę, że podejmujemy decyzję długofalową, której pokłosie zbierać będziemy przez lata. Inwestycja w technologię informatyczną stanowi potężny i odczuwalny w budżecie organizacji wydatek, który właściwie ulokowany przyniesie spodziewane zyski.

Rozpatrując dwa modele systemów przetwarzania danych - centralny i rozproszony, należy przede wszystkim dobrać taki, który spełniałby wymagania, jakimi są: elastyczność i efektywność. Dobór technologii przetwarzania zależy głównie od zadań, jakie ma ona spełniać, chociaż nie zawsze zadania stają się wyznacznikiem dla określonej implementacji. Duże znaczenie mają preferencje firm autorskich, decydujących o ostatecznym kształcie komputeryzacji. Związane jest to z oferowanymi na rynku produktami oraz doświadczeniem w stosowaniu określonych technologii.

Porównajmy technologie

Wybór określonej technologii przetwarzania danych należałoby rozpocząć od analizy wad i zalet każdej z nich. Pewne kwestie dotyczące atrybutów wskazanego modelu przetwarzania można przyjąć bez dyskusji, inne znowu wymagają szerokiego rozeznania kontekstu zastosowań. W literaturze dotyczącej systemów informatycznych [1,7,9] zwykło się powielać pewien szablon wad i zalet, jakby technika dostępu do zasobów danych i metod ich przetwarzania stanęła w miejscu, co dotyczy przede wszystkim przetwarzania centralnego, widzianego głównie jako tradycyjny układ komputera centralnego i dedykowanych terminali. Rozszerzając jednak pojęcie centralizmu, jako dostęp do wspólnych fizycznych zasobów danych (także przy użyciu architektury klient-serwer) zlokalizowanych w jednym miejscu i zarządzanych przez oprogramowanie serwera bazy danych oraz przy możliwościach rozbudowy punktu centralnego przetwarzania o kilka równolegle pracujących i zarządzających przetwarzaniem mainframe'ów, zauważamy zmiany w charakterystyce - jak się wówczas okazuje, pewne wady przestają istnieć. Tak więc przetwarzanie centralne możemy rozpatrywać w aspekcie tradycyjnym bądź nowoczesnym i od tego uzależniać kwalifikację jakości systemu. W poniższej charakterystyce porównawczej cechy odnoszące się do technologii przetwarzania centralnego w ujęciu tradycyjnym oznaczono znakiem zapytania jako dyskusyjne w kontekście dalej przytoczonych danych empirycznych.

Zaletami przetwarzania centralnego są:

- lepsza kontrola przetwarzania;
- wysoka wydajność dla przetwarzania transakcyjnego;
- łatwiejsza administracja systemem;
- redukcja powielanych danych i procedur zarządzających;
- lepsza kontrola wydatków na przetwarzanie danych;
- standaryzacja;
- wydajniejsze zarządzanie, koordynacja i rozwiązywanie problemów;
- (?) jednolitość narzędzi i aplikacji;
- (?) wspólna płaszczyzna sieciowa;
- (?) ustaleni użytkownicy i technologia.

Jako podstawowe wady przetwarzania centralnego należy wymienić:

- wysoki koszt początkowy;
- wymagany wysoko przeszkolony personel;
- brak kontroli użytkownika nad środowiskiem przetwarzania;
- mała elastyczność dopasowania do potrzeb indywidualnych;
- (?) mała uniwersalność;
- (?) brak możliwości lokalnego przetwarzania (pełne uzależnienie od centrali);
- (?) wysoki koszt połączeń teletransmisyjnych;
- (?) wysokie koszty związane z utrzymaniem systemu centralnego;
- (?) trudności rozbudowy;
- (?) unieruchomienie całego systemu przy awarii komputera centralnego;
- (?) spadek wydajności przy wzroście liczby jednoczesnych użytkowników i ich rozproszeniu geograficznym.

Rozpatrując pod tym samym kątem przetwarzanie rozproszone, zauważamy, że posiada ono niewątpliwe zalety, jakimi są:

- rozproszenie danych zgodnie z wymaganiami celów biznesowych;
- szybszy dostęp do informacji - użytkownik końcowy, pracując jedynie z podzbiorem informacji, najczęściej zgromadzonym w lokalnym serwerze, ma z natury rzeczy szybszy do nich dostęp niż do zasobów w modelu centralnym;
- szybsze przetwarzanie - system rozproszonych baz danych umożliwia przetwarzanie danych w różnych miejscach, optymalizując wykorzystanie zasobów komputerowych;
- możliwość rozbudowy - dodawanie nowych węzłów odbywa się bez zaburzeń dla istniejącej struktury przetwarzania, co umożliwia nadążanie za ekspansją geograficzną organizacji;
- ułatwiona komunikacja - oddziały lokalne organizacji, będąc jednostkami mniejszymi i znajdującymi się bliżej klienta, mają możliwość lepszego wykorzystania mechanizmów bezpośredniego kontaktu z użytkownikiem w powiązaniu z wydajniejszym dostępem do informacji;
- obniżenie kosztów - dużo bardziej efektywne jest podłączenie stacji roboczej do sieci niż dokonywanie rozbudowy centralnego komputera i dedykowanych linii teletransmisyjnych;
- sprzyjający interfejs - stacje robocze wyposażone zazwyczaj w interfejs graficzny są łatwiejsze w obsłudze i lepiej rozumiane przez użytkowników niedoświadczonych w technologiach komputerowych, przyspieszając tym samym proces adaptacji do nowego środowiska pracy;
- odporność na awarie - o ile w modelu centralnym uszkodzenie głównego systemu powoduje załamanie się całego systemu informatycznego, o tyle w przetwarzaniu rozproszonym zjawisko takie nie występuje; wskutek rozproszenia danych i ich przetwarzania awaria jednego węzła nie wpływa na pracę pozostałych;
- niezależność sprzętowa - użytkownik końcowy ma dostęp do aktualnej kopii danych, które mogą być przetwarzane na dowolnym komputerze; ukryta przed nim jest struktura sprzętu i środowisko systemowe.

Oprócz wielu zalet przetwarzanie rozproszone ma także swoje wady, do których zaliczyć należy:

- złożoność zarządzania - kontrola i zarządzanie rozproszonymi zasobami jest o wiele bardziej skomplikowane niż w wypadku modelu centralnego. Aplikacje muszą rozpoznawać położenie danych i łączyć je logicznie. Bardziej skomplikowane stają się procedury obsługi transakcji, kontroli dostępu, archiwacji i odtwarzania danych. Problemem staje się optymalizacja zapytań i optymalizacja dostępu do danych;
- bezpieczeństwo - im dane bardziej zdecentralizowane, tym problem zachowania ich jednoznaczności i nienaruszalności staje się trudniejszy. Biorąc pod uwagę, że zarządzanie danymi rozkłada się na lokalne węzły sieci, gdzie są one przechowywane i administrowane przez niezależne zespoły ludzi, do sprawy dochodzą trudności natury organizacyjnej; ciągle też jeszcze próg bezpieczeństwa sieci lokalnych pozostawia wiele do życzenia;
- brak standardów - przetwarzanie rozproszone bazuje na rozwiązaniach sieciowych, dla których ciągle nie istnieją standardy na poziomie protokołów transmisji; wykorzystywane do tej pory protokoły nie pozwalają na pełne rozwinięcie możliwości oferowanych przez przetwarzanie rozproszone.

Wskaźniki opłacalności

Obliczanie stopnia opłacalności inwestycji w system komputerowy nie jest zadaniem trywialnym, bowiem oprócz wartości mierzalnych, pozwalających w sposób dosyć dokładny przedstawić zależność kosztów - korzyści, pojawia się wiele czynników trudnych do oszacowania lub wręcz posiadających tło pozaekonomiczne. Etap ten zazwyczaj powoduje zaostrzenie kontrastu pomiędzy cechami funkcjonalnymi a kosztami systemu. Sprzeczność jest o tyle trudna do rozwiązania, że systemy o wysokich walorach użytkowo-funkcjonalnych oferowane są za wyższą cenę aniżeli konkurencyjne systemy, nie spełniające wymagań na należytym poziomie. Analiza kosztów - korzyści może być pomocna w kreowaniu polityki inwestycyjnej. Ocena kosztów systemu nigdy nie jest stuprocentowo dokładna, jako że wpływ na nią ma wiele czynników, do których należą:

- koszty bezpośrednie i pośrednie;
- wartości mierzalne i niemierzalne;
- koszty stałe i zmienne.

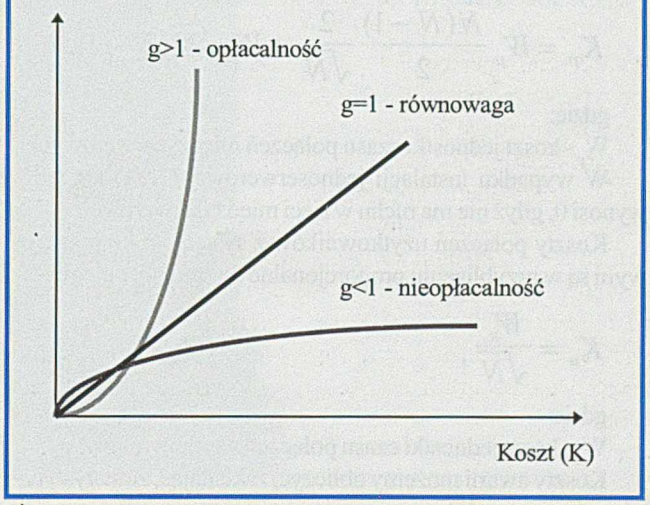
Obliczając stopień ekonomicznej opłacalności instalacji sprzętowej, można posłużyć się znanym wzorem Grosha [4], wyrażającym zależność mocy obliczeniowej komputera (M) i kosztów (K):

$$M = WK^g ,$$

gdzie W i g są współczynnikami.

System komputerowy spełnia założenia opłacalności ekonomicznej, jeżeli jego moc obliczeniowa wzrasta szybciej aniżeli koszty z tym związane.

Oszacowanie wskaźnika opłacalności



Źródło: na podstawie [4]

Waga ekonomicznej opłacalności wzrasta wraz ze złożonością systemu - ilością obsługiwanych użytkowników i przetwarzanych informacji oraz ich rozproszeniem. Tak więc moc obliczeniowa komputera jest tylko jednym ze składników świadczących o wydajności i opłacalności inwestycji. W dalszym jednak ciągu zależność ta nie oddaje znaczenia pozostałych

składników, ani tym bardziej nie wskazuje na preferowany model przetwarzania: centralny lub rozproszony.

Rozpatrując przypadek systemu dla dużej organizacji, należałoby obliczyć liczbę węzłów przetwarzania N, dającą maksimum wydajności przy najkorzystniejszej cenie. Zakładając, że każdy węzeł systemu jest identycznym pod względem parametrów wydajności i ceny systemem komputerowym, połączonym siecią każdy z każdym i obsługujący stałą liczbę użytkowników równomiernie rozproszonych, można wyrazić wzorem ogólną wartość kosztów (K) jako sumę kosztów sprzętu (K_s), połączeń między systemami (K_p), połączeń użytkowników z serwerem (K_u) oraz kosztów awarii (K_a).

$$K = K_s + K_p + K_u + K_a$$

Rozpatrzmy algorytmy obliczania poszczególnych składników kosztów.

Przyjmując zgodnie z prawem Grosha współczynnik g=2, koszt jednostki sprzętu komputerowego obliczymy według wzoru:

$$K_s = N\sqrt{M/WN} = W_s\sqrt{N} \quad ,$$

gdzie:

W_s - koszt jednostki czasu sprzętu.

Koszty połączeń sieciowych między serwerami obliczamy, mnożąc liczbę połączeń N(N-1)/2 przez koszt połączenia zależny od wielkości obciążenia (ruchu) sieci. Obciążenie połączenia otrzymamy, dzieląc ogólny ruch, który jest proporcjonalny do N-1, albowiem każda modyfikacja jest replikowana na N-1 pozostałych systemach, przez liczbę połączeń. Ruch przypadający na jedno połączenie jest odwrotnie proporcjonalny do N. Jeżeli przyjmiemy zależność pomiędzy ruchem w sieci a kosztem jako kwadratową, reprezentującą zgodnie z prawem Grosha opłacalność ekonomiczną, otrzymamy:

$$K_p = W_p \frac{N(N-1)}{2} \frac{2}{\sqrt{N}} = W_p\sqrt{N}(N-1),$$

gdzie:

W_p - koszt jednostki czasu połączeń międzyserwerowych.

W wypadku instalacji jednoserwerowej (N=1) koszt K_p wynosi 0, gdyż nie ma ruchu w sieci międzyserwerowej.

Koszty połączeń użytkowników z systemem komputerowym są w przybliżeniu proporcjonalne do średniej odległości:

$$K_u = \frac{W_u}{\sqrt{N}},$$

gdzie:

W_u - koszt jednostki czasu połączeń użytkowników.

Koszty awarii możemy obliczyć, zakładając, że statystycznie są one niezależne, a system jest uznawany za nieczynny w wypadku dysfunkcji wszystkich jego węzłów.

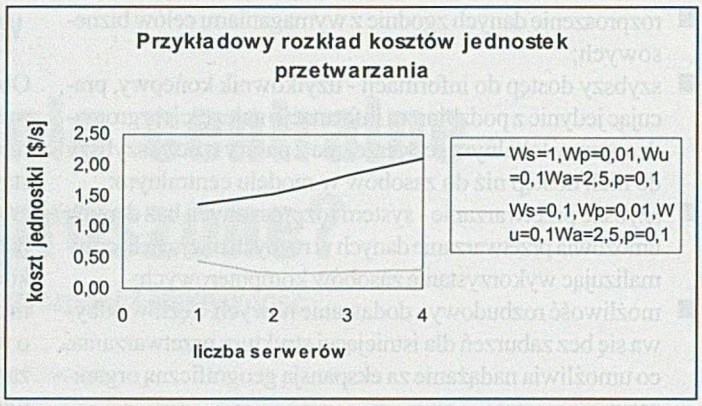
$$K_a = W_a p^N,$$

gdzie:

W_a - koszt jednostki czasu awarii;

p - prawdopodobieństwo (jednakowe) awarii każdego z węzłów systemu.

Model przyjęty do obliczeń jest bardzo uproszczony w stosunku do niejednorodnych modeli stosowanych w praktyce,



Źródło: opracowanie własne

aczkolwiek analiza pewnych zależności może uwypuklić istotne aspekty niektórych rozwiązań. Jeżeli ceny jednostek sprzętowych serwerów są bardzo wysokie a połączeń sieciowych niskie, wówczas powiększanie liczby węzłów powoduje istotny wzrost cenowy całości. Przy pewnych natomiast relacjach cenowych daje się uzyskać wyraźne obniżenie kosztu jednostkowego dla określonej liczby węzłów. Może także okazać się, że koszty awaryjności dla zastosowań strategicznych są niewspółmiernie wysokie w stosunku do cen jednostkowych sprzętu i połączeń sieciowych, co wymaga uwzględnienia przede wszystkim warunków bezpieczeństwa, a dopiero później optymalnej cenowo instalacji.

Dopasować koszt do wydajności

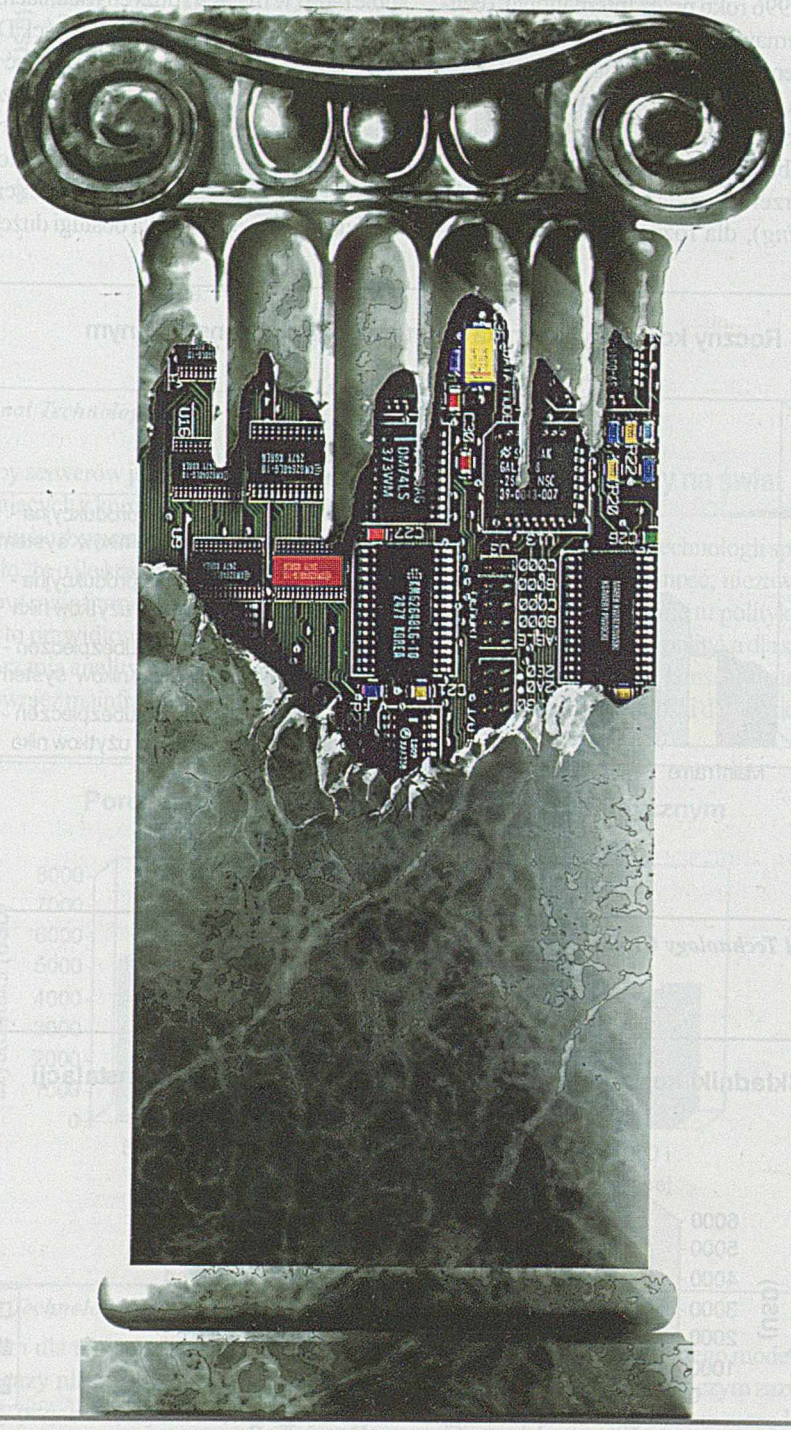
Ostatnie rozwiązania wprowadzone przez firmę IBM w systemie S/390 wyraźnie poprawiły wskaźnik kosztów - korzyści przetwarzania centralnego. W związku z rozpowszechnieniem się technologii CMOS (*Complementary Metal Oxide Semiconductor*) cena za MIPS spadła z 23250 USD w 1995 roku do poziomu 13780 USD w 1996 roku. Według danych opublikowanych przez International Technology Group, średni roczny koszt przypadający na użytkownika komputera mainframe wynosi 5183 USD, podczas gdy dla użytkownika platformy UNIX, 7947 USD rocznie - dla niewielkich instalacji do 100 użytkowników i dysproporcja ta zwiększa się wyraźnie przy dużych instalacjach, ponad 1000 użytkowników, dając koszty odpowiednio 1225 USD dla mainframe i 4170 USD dla serwerów UNIX-a. Wyraźnie też zwiększa się zainteresowanie instalacjami systemów opartych na komputerach mainframe S/390, które po odnotowanym spadku w 1994 roku obecnie wykazują tendencje zwyżkujące.

Raport ITG został oparty na badaniach przeprowadzonych w USA w 212 instytucjach, stosujących ogółem 273 instalacje systemów mainframe i UNIX (w układzie centralnym i rozproszonym), w tym 219 instalacji mainframe, 14 instalacji UNIX z serwerami MPP i 474 z SMP lub jednoprocesorowymi.

Koszty użytkownika były obliczane na podstawie pięcioletniego poziomu kosztów przypadających na sprzęt (zakup i utrzymanie), oprogramowanie systemowe (systemy operacyjne, podsystemy, systemy zarządzania bazami danych, oprogramowanie narzędziowe), koszty personelu informatycznego.

Natomiast nie były brane pod uwagę koszty: oprogramowania użytkowego (zakupu, projektowania i utrzymania), prze-

UNOWOCZEŚNIAMY
TWÓJ BANK



SOFTBANK® S.A.

BANKOWOŚĆ I SYSTEMY BANKOWE

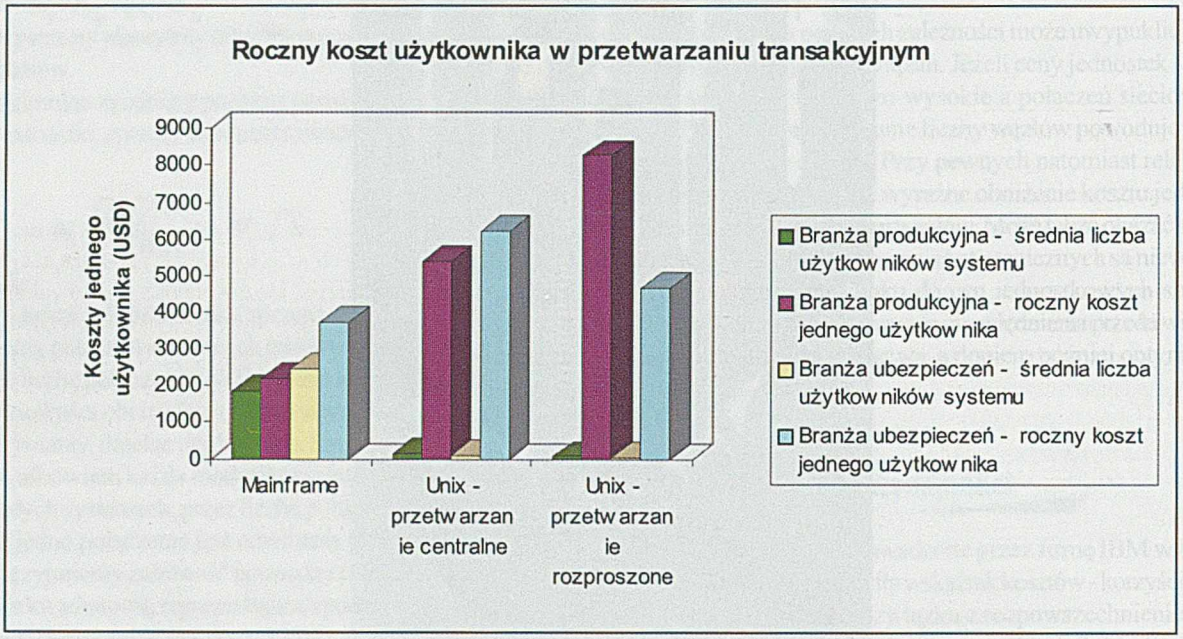
SOFTBANK S.A. ul.17-Stycznia 72A, 02-146 Warszawa, tel. 846 54 71, fax 846 14 57 .

twarzania użytkownika końcowego (komputery personalne, sieci lokalne), połączeń sieciowych, zasilania oraz wszelkie koszty związane z wdrażaniem i uruchamianiem systemu lub aplikacji.

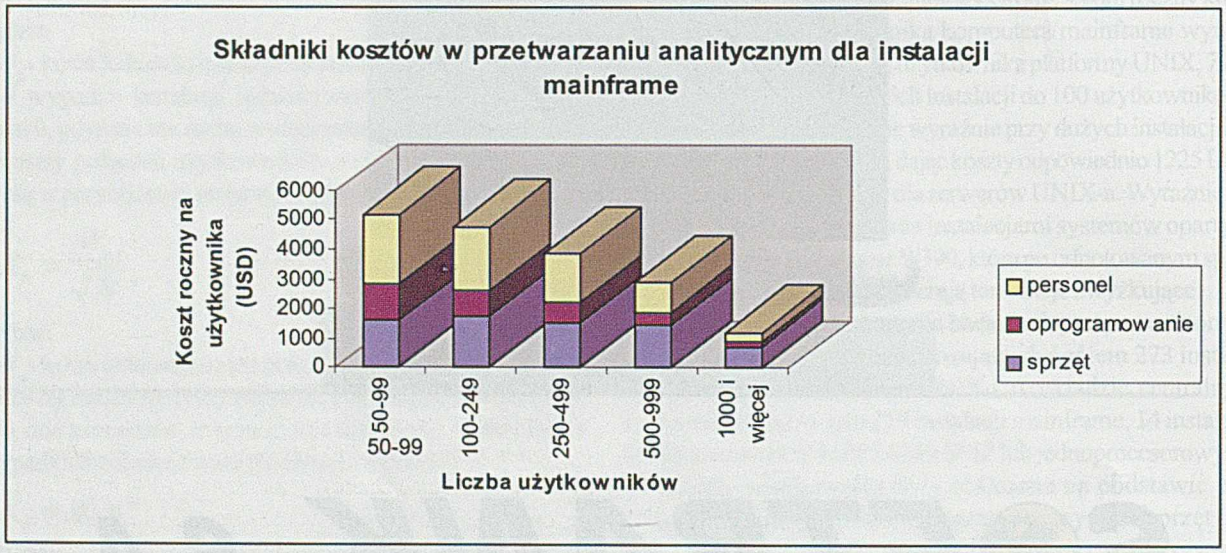
Dane opublikowane w 1996 roku przez International Technology Group wyraźnie przemawiają na korzyść przetwarzania centralnego z wykorzystaniem komputerów klasy mainframe - parametry, jakimi są: średnia liczba użytkowników pojedynczego systemu przy relatywnie niskich kosztach, zdecydowanie wygrywają z rozwiązaniami bazującymi na UNIX-ie.

Dotyczy to zarówno przetwarzania transakcyjnego, jak i analitycznego (*data mining*), dla różnych branży stosujących owe rozwiązania (produkcja, ubezpieczenia, dystrybucja, finanse).

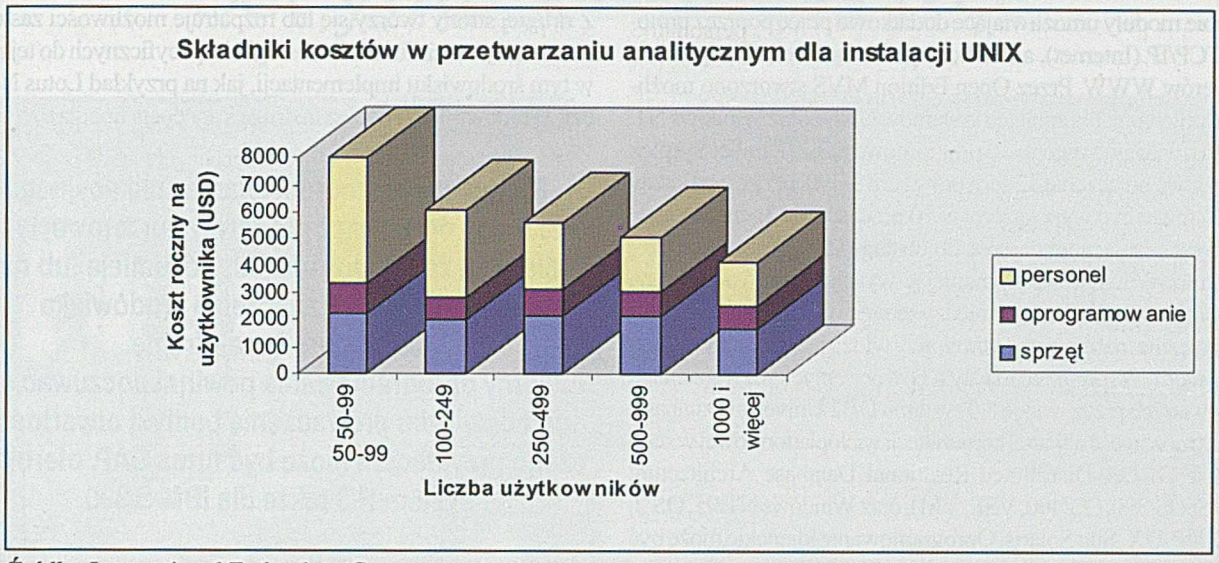
Analiza wykazuje, iż czynnikiem różnicującym systemy mainframe i unixowe jest liczba jednocześnie obsługiwanych użytkowników, liczba przetwarzanych zadań, wielkość stosowanych aplikacji - wszystkie parametry oczywiście dużo korzystniej przedstawiają się dla „dużych" maszyn. Nie bez znaczenia jest tu fakt lepszej skalowalności mainframe'ów, które z powodzeniem dają sobie radę i w małych i dużych zadaniach, nie naruszając podstawowego wskaźnika kosztów-korzyści. Duża moc obliczeniowa komputerów mainframe pozwala unikać w większości przypadków zjawiska dublowania serwerów, co zdarza się w instalacjach unixowych, zwłaszcza w wypadku przetwarzania rozproszonego, gdzie pociągnięcia takie trzeba wykonywać, chcąc uzyskać odpowiednią wydajność. Pojedyncze serwery SMP nie zapewniają dostatecznej mocy dla obsługi dużej liczby użytkowników oraz mnogości wykonywanych zadań, w związku z czym wymagają tworzenia klastrów. Serwery MPP, mimo iż wykazują lepszy wskaźnik skalowalności i mniejsze koszty roczne użytkownika, nie są także w stanie dorównać mainframe'om. Ujemnym zjawi-



Źródło: International Technology Group



Źródło: International Technology Group



Źródło: International Technology Group

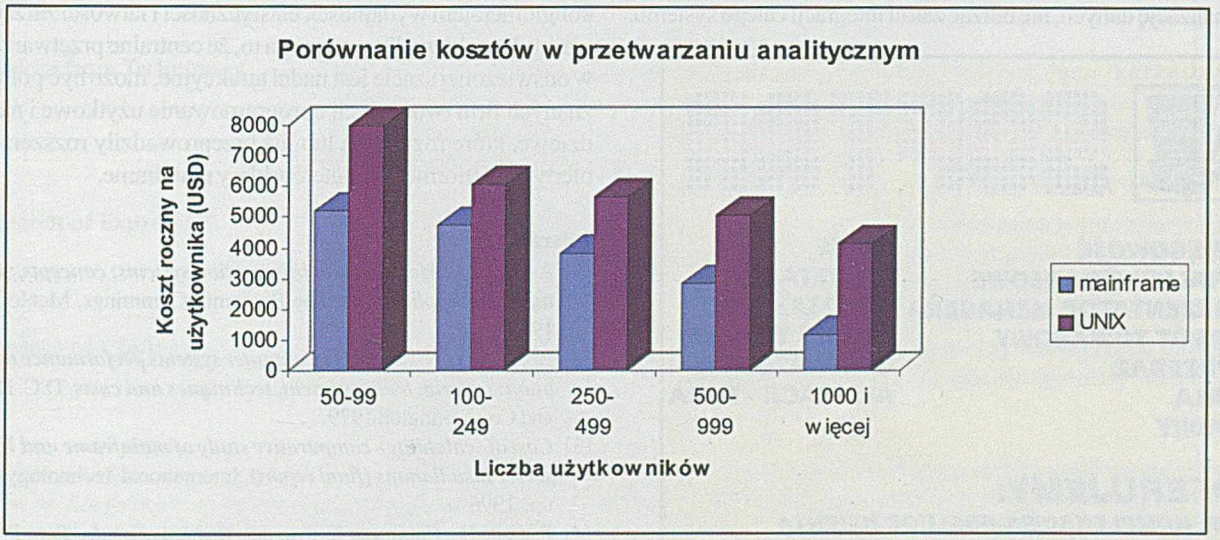
skiem pomnażania liczby serwerów jest napędzanie spirali kosztów pośrednich, wynikających z konieczności powielania oprogramowania oraz wzrostu liczby personelu informatycznego.

Zdawać by się mogło, że o ile koszty przetwarzania transakcyjnego dla komputerów mainframe są niższe niż w wypadku systemów unixowych, to prawidłowość ta nie zostanie zachowana dla trybu przetwarzania analitycznego (*query-intensive*). Lecz w tym wypadku również mainframe'y wyprzedzają konkurencję: koszt użytkownika dla małych instalacji, od 50 do 99 użytkowników, jest 1,5 razy niższy niż dla systemów unixowych i aż 3,4 razy niższy przy instalacjach obsługujących ponad 1000 użytkowników.

Planując nową instalację systemu komputerowego, trzeba bardzo dokładnie rozważyć wybór platformy sprzętowej, który o tyle jest łatwy, że dla każdej instalacji, obsługującej ponad 50 użytkowników, wszystko zdaje się przemawiać na korzyść komputerów mainframe – i przemawia tym bardziej, im liczba ta rośnie

Mainframe otwarty na świat

Oprócz doskonalenia technologii sprzętowej obniżającej koszty a zwiększającej wydajność, niezawodność i skalowalność, nie bez znaczenia wydaje się tu polityka prowadzona w dziedzinie oprogramowania. O ile polityka dla systemów przetwarzania rozproszonego była od lat konsekwentnie prowadzona w kierunku skalowalności i otwartości dla użytkownika, o tyle systemy pracujące w trybie transakcyjnego modelu host - terminal wydawały się niereformowalne. To z czym zazwyczaj kojarzono duże systemy komputerowe, to system zarządzania bazami danych DB2 współpracujący z transakcyjnym systemem CICS (*Customer Information Control System*) lub IMS (*Information Management System*) - oferujące znakowy interfejs i trudno modyfikowalne zastosowania, nie uwzględniające indywidualnych potrzeb użytkowników. Mainframe'y nie zawężają obecnie swego pola działania do sztywnej z punktu widzenia użytkownika technologii przetwarzania. Duża otwartość umożliwia wykonanie tego samego, co w ogólnie lubianym środowisku graficznym, z tym, że na komputerze wyposażonym w ogromne możliwości. Następca systemu operacyjnego MVS OS/390 to produkt zawierający



Źródło: International Technology Group

w sobie moduły umożliwiające dodatkowo pracę poprzez protokół TCP/IP (Internet), architekturę klient-serwer, obsługiwanie serwerów WWW. Przez Open Edition MVS stworzono możliwość uruchamiania aplikacji systemów UNIX oraz Windows NT.

Rozwiązanie sprzętowo programowe S/390 Parallel Sysplex umożliwia połączenie 32 serwerów serii S/390, przy czym każdy z nich może być wyposażony w 10 procesorów. Technologia ta ma wystarczającą wydajność do obsługi 150 tysięcy jednoczesnych użytkowników, zapewniając wysoką niezawodność, na poziomie 99,99%. Praca pomiędzy serwerami rozdzielana jest automatycznie, natomiast użytkownicy widzą jeden spójny logicznie system. Wspomnieć należy o nowoczesnych rozwiązaniach oferowanych przez wersję 5.0 systemu DB2 Universal Database, dającą możliwość otwartej komunikacji wieloplatformowej w standardzie DRDA-Distributed Relational Database Architecture (MVS, OS/390, OS/400, VSE, VM), oraz Windows NT/95, OS/2, AIX, HP-UX, Sun Solaris. Oprogramowanie klienckie może być tworzone na wymienionych platformach i dodatkowo dla systemów: DOS, MacOs, IRIX (*Silicon Graphics*). Dostępność przez interfejs Java i JDBC (*Java Database Connectivity*) czyni środowisko pracy rzeczywiście otwarte, dostosowane zarówno do przetwarzania transakcyjnego, jak i analitycznego. Umożliwia się tym samym konfigurowanie złożonych instalacji heterogenicznych, w których jako komputery back-end stosowane są maszyny typu mainframe, natomiast front-end stanowią serwery klasy midrange, operujące na płaszczyźnie jednej z odmian systemów UNIX lub Windows NT. Mechanizmy replikacji danych pozwalają na rzeczywiste rozproszenie środowiska przetwarzania i regionalizację danych, nie burząc zasad integracji całego systemu.

Z drugiej strony tworzy się lub rozpatruje możliwości zastosowań maszyn mainframe do obsługi niespecyficznych do tej pory w tym środowisku implementacji, jak na przykład Lotus Notes czy Windows NT (chociaż ta druga koncepcja upadła).

Producenci oprogramowania aplikacyjnego nie mogą ograniczać platformy sprzętowej tylko dlatego, że nie potrafią, nie rozumieją lub nie chcą zauważyć znaczenia środowiska komputerów mainframe. Twórcy oprogramowania powinni poczuwać się do obowiązku prowadzenia polityki otwartości, czego przykładem może być firma SAP, oferująca system R/3 także dla IBM S/390.

Jak z przytoczonych faktów można wywnioskować, przetwarzanie centralne było hermetyczne w przeszłości. Dzisiaj rozwija się ono w kierunku, dającym użytkownikom rozwiązania na poziomie przetwarzania rozproszonego, ale podwyższając niezawodność, ułatwiając zarządzanie systemem i oferując zdecydowanie niższe koszty. Wydaje się, że ta metoda połączenia zalet obydwu modeli: centralnego i rozproszonego jest przy dzisiejszych możliwościach rozwiązaniem optymalnym. Przechwytywanie pozytywnych cech centralizmu jest manifestowane zresztą dosyć wyraźnie w dziedzinie komputerów osobistych, gdzie próbuje się opracować nowy model komputera sieciowego, będącego konglomeratem wydajności, elastyczności i łatwości zarządzania środowiskiem. Dowodem na to, że centralne przetwarzanie w odświeżonej szacie jest nadal atrakcyjne, może być polityka znanych firm tworzących oprogramowanie użytkowe i narzędziowe, które rozpatrują lub już przeprowadziły rozszerzenie oferty na platformę komputerów klasy mainframe.

Literatura:

[1] Awad E.M., ***Management information systems: concepts, structure, and applications***, The Benjamin Cummings, Menlo Park 1988.

[2] Borovits I., Neumann S., ***Computer systems performance evaluation: criteria, measurement, techniques and costs***, D.C. Heath and Co., Lexington 1979.

[3] ***Cost of scalability - comparative study of mainframe and UNIX server installations (final report)***, International Technology Group, 1996.

[4] Ferrari D., ***Computer systems performance evaluation***, Prentice-Hall Inc., Englewood Cliffs 1978.

[5] Kroenke D., ***Management information systems***, Mitchell Publishing, Santa Cruz 1989.

[6] McGee R. W., ***Accounting for data processing costs***, Quorum Books, New York 1988.

[7] Rob P., Coronel C., ***Database systems: design, implementation and management***, Boyd and Fraset, Danvers 1995.

[8] Simpson D., ***Are mainframes cool again?***, Datamation, April 1997.

[9] Stamper D., Price W., ***Database design & management: an applied approach***, McGraw-Hill, New York 1990.

Mgr inż. **Piotr Kowalski** jest kierownikiem Pracowni Obsługi Informatycznej Bibliotek Uniwersytetu Śląskiego oraz głównym specjalistą ds. informatyki w Górnośląskiej Wyższej Szkole Handlowej w Katowicach.

Set your business free

Freedom from Technology.

IFS Applications - informatyczne rozwiązania wspierające zarządzanie przedsiębiorstwem - wykorzystują standardową metodologię oraz narzędzia ogólnie dostępne na rynku. Dzięki temu nie uzależniasz się od rzadko stosowanej technologii informatycznej.

Freedom of Expression.

IFS Applications dostępne są na wiodących otwartych platformach informatycznych takich, jak: Windows NT, UNIX oraz Open VMS. Możesz z łatwością korzystać z funkcji graficznych, tekstowych, video i dźwięku, JAVA oraz telefonii komórkowej.

Freedom of Choice.

Modularna architektura IFS Applications umożliwia dowolną konfigurację ponad 30 standardowych modułów rozwijanych na bieżąco tak, by zawsze spełniały wymagania konkretnego przedsiębiorstwa.

Od 1983 roku tworzymy informatyczne rozwiązania wspierające zarządzanie przedsiębiorstwem. Pracując nad IFS Applications opieramy się na bardzo prostych zasadach: Wierzymy, że przedsiębiorstwa, tak jak i ludzie, chcą mieć możliwość nieograniczonego rozwoju. Chcą swobody. Chcą rosnąć i zmieniać się. Chcą to robić tak, by nie uzależniać się od skomplikowanej technologii informatycznej. IFS Applications umożliwia Twojemu przedsiębiorstwu swobodny, nieograniczony rozwój, którego celem jest osiągnięcie i utrzymanie wiodącej pozycji na rynku.

Gdziekolwiek jesteś, niezależnie od pory dnia, specjaliści IFS są zawsze do Twojej dyspozycji. Międzynarodowa organizacja IFS zapewnia swoim klientom wsparcie na całym świecie. Zaufali nam już min. *Akzo Nobel, Caterpillar, International Paper, Ericsson, NEC, VOLVO...*

Dołącz do Elity.

Industrial & Financial Systems

IFS

Warszawa
Tel. (48 22) 608 46 00
Fax (48 22) 608 46 01

Kraków
Tel. (48 12) 422 50 15
Fax (48 12) 421 45 29

Gdańsk
Tel. (48 58) 345 29 64
Fax (48 58) 345 29 67

Poznań
Tel. (48 61) 851 92 55
Fax (48 61) 851 96 50

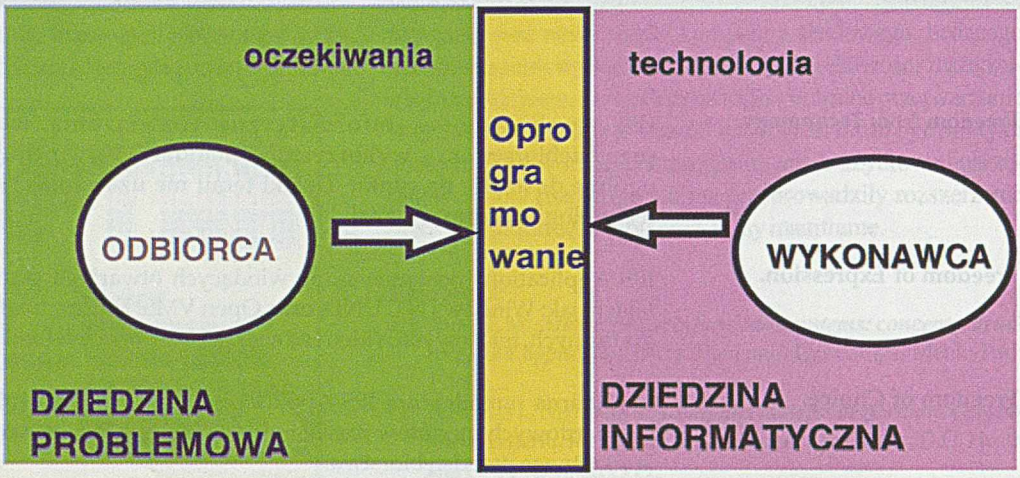
http://www.ifsab.com info@ifs.com.pl

Inżynieria wymagań,
czyli jak unikać zbędnej pracy

Janusz Górski

Informatyka w coraz większym stopniu staje się dziedziną uniwersalną.
Stykamy się z nią w nieomal każdej sferze działalności, w pracy i podczas odpoczynku, w podróży, w nauce czy podczas korzystania z różnych usług.

Nasza świadomość informatyczna ma wiele źródeł, takich jak lektura książek, doniesienia prasowe czy TV, filmy, obserwacje w trakcie wizyt w wysoko rozwiniętych krajach, ale również, coraz częściej, doświadczenia bezpośrednie, związane z informatyzacją obsługi w kasach kolejowych, bankach, urzędach czy sklepach, z wykorzystaniem skomputeryzowanego sprzętu audio video czy wreszcie własne doświadczenia z komputerem w pracy lub w domu. W coraz większym stopniu zaczynamy również rozumieć, że jakość naszego życia zależy od udanych zastosowań informatyki oraz związanych z tym kosztów. I ta tendencja będzie się z pewnością nasilać. Warto więc zastanowić się nad uwarunkowaniami sukcesu przedsięwzięcia polegającego na zastosowaniu technologii informatycznej do rozwiązania problemu z otaczającego nas świata.

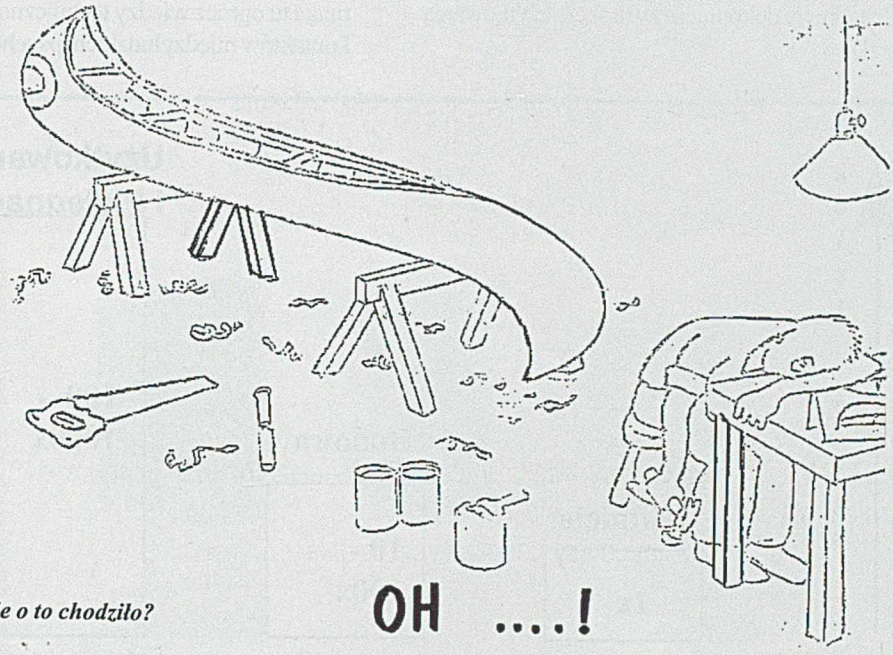


Rys. 1. Zdecydowana większość użytecznego oprogramowania powstaje w układzie klient-wykonawca

Najpierw „co?", potem „jak?"

W tym artykule założymy, że interesują nas przedsięwzięcia informatyczne, w których zaangażowane są dwie strony: ***odbiorca*** - który jest źródłem zapotrzebowania na system informatyczny oraz ***wykonawca*** - odpowiedzialny za realizację tego systemu. Związek pomiędzy tymi stronami jest ustalany przez zawarcie ***kontraktu*** na wykonanie systemu. Rzeczywiste sytuacje mogą być oczywiście bardziej skomplikowane. Po stronie odbiorcy można wyróżnić klienta, który finansuje system, użytkowników systemu oraz jego obsługę techniczną. Po stronie wykonawcy sytuacja może być również bardziej złożona - można tu wyróżnić dostawców sprzętu i podstawowego oprogramowania, projektantów i realizatorów, wdrożeniowców oraz dostawcę końcowej wersji systemu.

Podstawowa różnica pomiędzy odbiorcą i wykonawcą, która będzie dla nas dalej istotna, polega na tym, że interes odbiorcy jest określany przede wszystkim przez to, ***czym będzie system informatyczny***, podczas gdy wykonawca reprezentuje ***sposób realizacji tego systemu***. Rozróżnienie to jest więc w swojej istocie rozróżnieniem pomiędzy ***identyfikacją*** a ***rozwiązaniem*** problemu związanego z danym przedsięwzięciem. Na bazie tego rozróżnienia możemy też wyeksponować jeden z najpoważniejszych problemów związanych z zastosowaniami informatyki: nawet najlepsze, w sensie technicznym, rozwiązanie źle zidentyfikowanego problemu nie będzie uznane za satysfakcjonujące. Innymi słowy, odbiorca nie będzie zadowolony z ***niepotrzebnego systemu***.

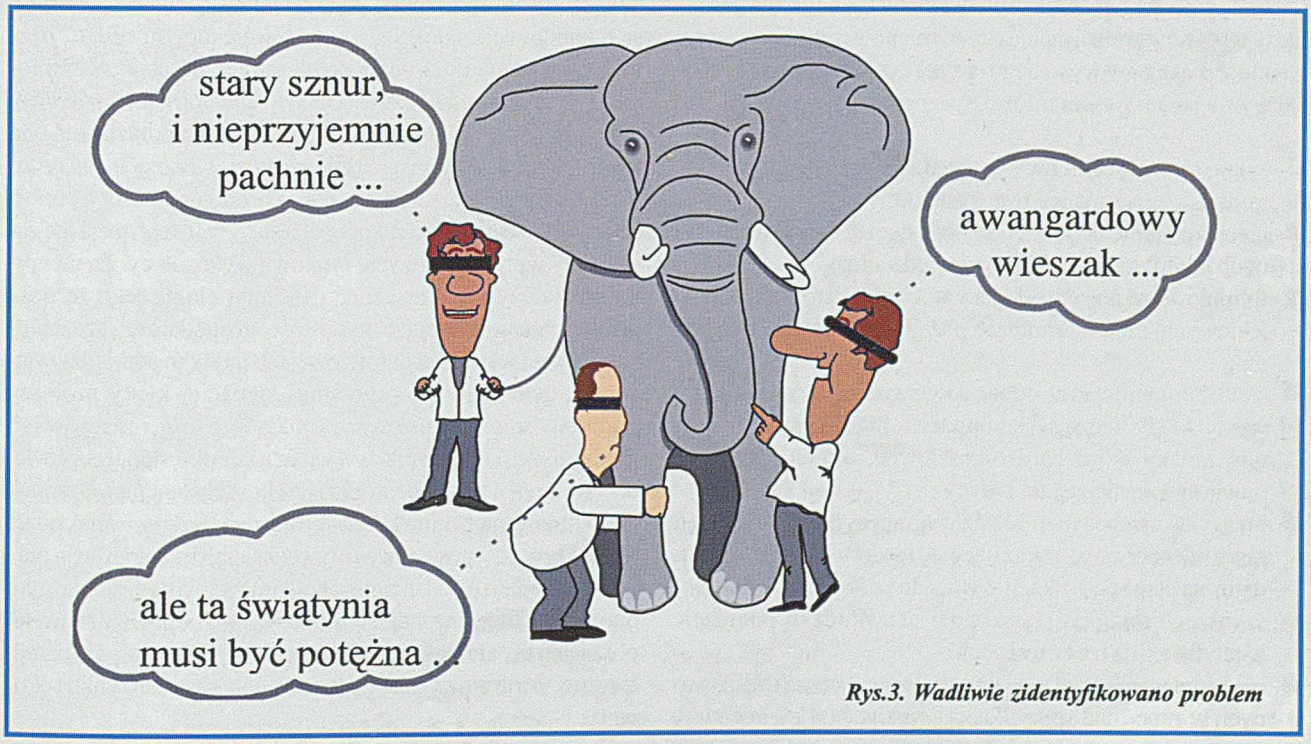


Rys. 2. Czy rzeczywiście o to chodziło?

Poprawianie jest zbyt kosztowne

Nawet najlepsza technologia wykonania systemu informatycznego nie gwarantuje więc sukcesu bez właściwej identyfikacji celu i zadań, które ten system ma zrealizować. Dlatego każdy projekt informatyczny rozpoczyna się etapem ***specyfikacji wymagań systemowych***. Celem tego etapu jest odpowiedź na pytanie, czym ma być budowany system, a jego produktem jest dokument określający wymagania dotyczące systemu. Kluczową rolę w identyfikacji tych wymagań odgrywają reprezentanci odbiorcy, gdyż tylko oni mogą odpowiedzieć na pytania dotyczące zakresu, funkcji, ograniczeń, horyzontu czasowego czy przewidywanych zmian związanych z systemem. Błędne odpowiedzi na powyższe pytania (lub ich brak) prowadzą zwykle do sytuacji, w której system różni się od potrzeb i oczekiwań odbiorcy.

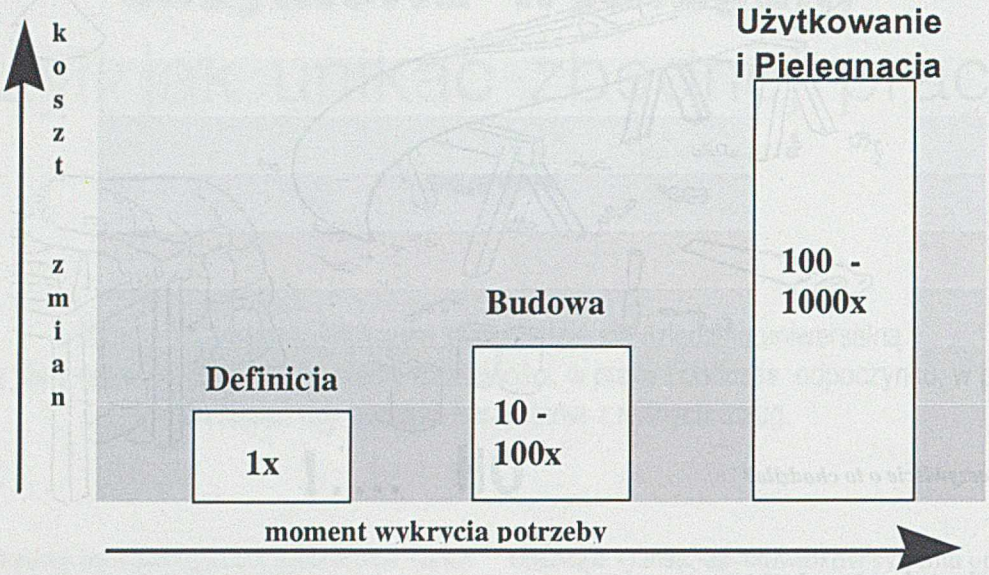
Pozornie można temu zaradzić na drodze zmian i modyfikacji, mających na celu „poprawienie" systemu w czasie jego budowy lub w trakcie użytkowania. Rozwiązanie takie jest jednak bardzo kosztowne. Dane zebrane na podstawie wielu projektów wskazują, że koszty naprawy defektów wymagań rosną w miarę postępu prac zgodnie z zasadą 1:10. Oznacza to, że naprawa defektu wymagań systemowych kosztuje 10 razy więcej, gdy jest on usuwany na etapie projektowania, 100 razy więcej, gdy dotyczy to etapu realizacji (implementacji), a 1000



Rys.3. Wadliwie zidentyfikowano problem

razy więcej, jeżeli naprawy dokonujemy w trakcie użytkowania systemu (por. rys.4).

maga tu oprócz wiedzy technicznej umiejętności z zakresu kontaktów międzyludzkich, psychologii i socjologii);



Rys.4. Zależność kosztu wprowadzania zmian od fazy projektu, w której nastąpił błąd

Przede wszystkim zastanów się, co chcesz zrobić...

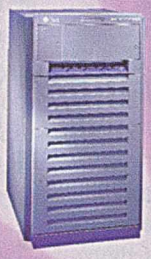
Z powyższych obserwacji wynika, że specyfikacji wymagań systemowych warto poświęcić szczególną uwagę. Unikanie lub wczesna eliminacja defektów specyfikacji umożliwia znaczne obniżenie kosztu i w zasadniczy sposób zwiększa szansę na sukces przedsięwzięcia informatycznego. Technologia związana z uzyskaniem i utrzymywaniem wysokiej jakości specyfikacji w ramach projektu informatycznego jest nazywana ***inżynierią wymagań***. Opanowanie tej technologii ma zasadniczy wpływ na zmniejszenie ryzyka niepowodzenia projektu. Niektóre przyczyny wysokiego ryzyka związane ze specyfikacją wymagań na system informatyczny, to:

- ciągła ewolucja środowiska docelowego systemu;
- trudność ustalenia zakresu systemu;
- nieuświadomienie przez odbiorcę swoich oczekiwań i potrzeb lub trudności w ich wypowiedzeniu;
- formułowanie potrzeb odbiorcy w terminach rozwiązań projektowych, a nie w terminach pożądanych własności systemu;
- uświadomienie przez odbiorców swoich potrzeb dopiero wtedy, kiedy widzą system podczas pracy (a więc wtedy, gdy zmiany są już bardzo kosztowne, zgodnie ze wspomnianą wcześniej regułą 1:10);
- brak właściwego zrozumienia wagi specyfikacji wymagań przez menedżerów, wyrażający się niedostatecznymi nakładami na etap specyfikacji wymagań i dążeniem do jak najszybszego podjęcia „konkretnej pracy" (czyli pomijanie identyfikacji na rzecz realizacji);
- niedostateczna wiedza i wyszkolenie osób zaangażowanych w prace nad specyfikacją wymagań (wiele zdań wy-
- pomijanie lub nieumiejętne podejmowanie działań mających na celu zapewnienie i utrzymanie jakości wytworzonych specyfikacji.

... i to się wszystkim opłaci

Wysoka jakość wymagań dotyczących systemu jest w interesie zarówno odbiorców, jak i wykonawców. Odbiorców, ponieważ ich celem jest system, który spełni oczekiwania i rozwiąże związane z nimi problemy. Wykonawców, ponieważ daje szansę na uniknięcie sytuacji „pogoni za ruchomym celem" oraz tworzy podstawę do sformułowania kryterium zakończenia projektu. Zakładamy tu oczywiście sytuację normalną, tzn. taką, w której nieudany system informatyczny jest porażką dla obu stron. Oznacza to, że projekt związany z wytworzeniem systemu wynika z rzeczywistych potrzeb odbiorcy, który ponosi koszty jego realizacji, a zakończona sukcesem realizacja ma zasadniczy wpływ na pozycję rynkową wykonawcy. Za nienormalne należy więc tu uznać (niestety, ciągle jeszcze dość powszechne w naszej rzeczywistości gospodarczej, szczególnie w sferze własności państwowej lub na styku pomiędzy własnością państwową i prywatną) sytuacje, kiedy projekty informatyczne są realizowane w układach grup interesów lub w warunkach skrajnej niekompetencji bez określonej odpowiedzialności za wydawane środki i bez konsekwencji związanych z podejmowaniem niewłaściwych decyzji. Należy jednak oczekiwać, że wraz z postępującą prywatyzacją, decentralizacją oraz usprawnianiem mechanizmów podejmowania decyzji i jasnym przypisaniem związanej z tymi decyzjami odpowiedzialności obszar ten będzie się kurczył i w coraz większym stopniu przedsięwzięcia informatyczne będą podlegały prawom rynku i ekonomii.

Enterprise 6000
Wysokiej wydajności i skalowalności serwer
dla bardzo dużych aplikacji baz danych.

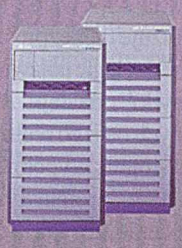


Enterprise 10000
Najwyższej wydajności serwer
spełniający zadania hurtowni danych.

Ultra Enterprise 450
Serwer dla grup roboczych
wykorzystujących technologie największych
systemów wieloprocesorowych.

Enterprise 2
Silny, wieloprocesorowy serwer
dla grup roboczych.

Enterprise 4000
Łatwo dostępny serwer o dużej
wydajności i skalowalności.



Enterprise Cluster HA
Serwer przeznaczony dla środowisk
biznesowych, gdzie pewność działania
i wydajność są najważniejsze.

Sun RSM Array 2000
Wysokiej wydajności macierz RAID
dla korporacyjnego centrum danych.

Rozwój wymaga wydajnych narzędzi

http://www.sun.com.pl

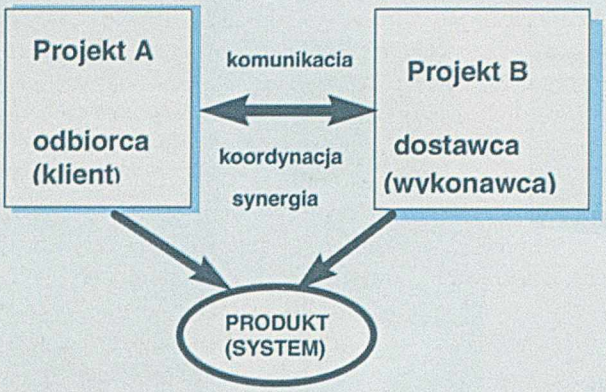
Praca w sieci komputerowej, która nie jest skalowalna po prostu się nie opłaca. Dlatego też firma Sun Microsystems oferuje pełną serię skalowalnych serwerów oraz systemów magazynowania danych, które bez kłopotów integrują sieci - począwszy od biurkowych pecetów, a na kompleksowych środowiskach komputerowych skończywszy. Pojedynczy system firmy Sun umożliwia zarządzanie całą korporacją, zapewniając bezawaryjność przy doskonałym stosunku ceny do wydajności. Nasze systemy zabezpieczeń gwarantują pewne i bezpieczne korzystanie z aplikacji wspomagania decyzji, zarządzania zasobami oraz systemów intranetowych. Środowisko operacyjne Solaris, lepiej niż inne systemy operacyjne, pozwala na uruchomienie rozbudowanych programów, obsługujących od jednego do tysięcy użytkowników. Nasze technologie oparte na języku programowania Java oferują płynne przejście do nowoczesnych systemów otwartych, znosząc wszelkie bariery pracy w sieci. Dlatego właśnie przedsiębiorstwa wybierają rozwiązania Suna - najlepsze pod słońcem!

Sun microsystems

Sun Microsystems Poland, Warszawa, ul.Hankiewicza 2, tel.: (48 22) 874 78 00, fax: (48 22) 658 40 60, e-mail: info@Poland.Sun.com

Jak się do tego zabrać?

Wymagania systemowe powstają w efekcie zaangażowania dwóch stron- odbiorcy i wykonawcy systemu informatycznego. Niedostateczny udział jednej ze stron zwiększa ryzyko błędów i związanych z nimi negatywnych skutków.



Rys.5. Przedsięwzięcie informatyczne to najczęściej dwa projekty, jeden po stronie odbiorcy i jeden po stronie wykonawcy

Obecny stan technologii w zakresie inżynierii wymagań w kraju należy uznać za wysoce niezadowalający (chociaż sytuacja na świecie też jest daleka od ideału). Wynika to z wielu powodów, takich jak braki edukacyjne (zagadnienia inżynierii wymagań czy szerzej inżynierii oprogramowania są rzadko obecne w programach nauczania naszych uczelni wyższych), brak ustabilizowanego rynku zastosowań informatyki i związanych z nim doświadczeń, wspomniane wcześniej rozmycie odpowiedzialności za błędne decyzje oraz różnego rodzaju praktyki protekcyjne chroniące „nieudaczników" przed działaniem mechanizmów rynku. Polepszenie sytuacji wymaga rozbudzenia świadomości potrzeby działań naprawczych oraz transferu technologii w zakresie inżynierii wymagań zarówno do sfery producentów systemów informatycznych, jak i do sfery użytkowników tych systemów. W uświadomieniu sobie stanu własnej instytucji w zakresie inżynierii wymagań może być pomocna próba odpowiedzi na następujące pytania, w odniesieniu do wybranego projektu informatycznego:

- Czy dokument specyfikacji wymagań stanowił załącznik do kontraktu pomiędzy odbiorcą i wykonawcą?
- Czy jego treść była utworzona na bazie zdefiniowanych wzorców (standardów), a proces wytworzenia był realizowany na bazie zdefiniowanych procedur postępowania?
- Czy wymagania były mierzalne w tym sensie, że fakt ich spełnienia mógł być rozstrzygnięty na bazie obiektywnego kryterium?
- Czy wymagania były kompletowane z uwzględnieniem jawnie zdefiniowanego zakresu systemu i z dostatecznym udziałem odbiorcy?
- Czy poddano analizie zgodność wymagań z celami strategicznymi instytucji odbiorcy?
- Czy przewidziano i zrealizowano zabiegi analityczne mające na celu zapewnienie i utrzymanie jakości wymagań?
- Czy zadbano o utrzymanie spójności wymagań z innymi reprezentacjami systemu (np. dokumentacją projektową, kodem źródłowym, dokumentacją testów)?
- Czy przewidziano i stosowano procedury postępowania związane z wprowadzaniem zmian do wymagań dotyczących systemu?

Negatywna odpowiedź na choćby jedno z powyższych pytań sygnalizuje niebezpieczeństwo, że projekt zostanie zdominowany przez ciągle pojawiające się zapotrzebowania na zmiany i „poprawki", nastąpi znaczne przekroczenie zasobów przeznaczonych na jego realizację oraz dojdzie do kontrowersji odnośnie do stopnia wypełnienia zobowiązań przewidzianych kontraktem (tzw. „projekt Syzyfa" - rys.6). Stanowi to również realne zagrożenie, że realizacja projektu będzie dotyczyła systemu, który nie odpowiada oczekiwaniom odbiorcy. Oczywiście, w celu określenia zakresu i kolejności posunięć zmierzających do zmiany stanu rzeczy niezbędna jest znacznie głębsza i bardziej wnikliwa analiza niż prosta odpowiedź na kilka przedstawionych powyżej pytań. Ocena taka daje podstawę do opracowania kolejnych etapów związanych z wdrożeniem technologii inżynierii wymagań w danej instytucji.

Rys.6. „Projekt Syzyfa", czyli pogoń za ruchomym celem

I potem trzeba jeszcze to wszystko pielęgnować!

Zatwierdzony dokument, który zawiera specyfikację wymagań, stanowi zwykle część kontraktu na budowę systemu. Na tej bazie wykonawca szacuje czas i koszt projektu. Kryje się tu jednak bardzo poważne niebezpieczeństwo. Wymagania, które są uzasadnione w czasie ich specyfikowania, podlegają zmianom wraz ze zmianą warunków, w jakich funkcjonują organizacje klienta i wykonawcy oraz w jakich powstaje system. Jeżeli zmiany te nie zostaną odpowiednio wcześnie rozpoznane i uwzględnione (również w kontrakcie), klient nie otrzyma oczekiwanego systemu, a wykonawca może mieć trudności z wyegzekwowaniem zapłaty za część wykonanej pracy. Jeżeli zaprzeczymy konieczności nadążania za zmianami lub je zignorujemy, „zamrażając" wymagania na zawsze, prawie na pewno powstanie niewłaściwy system.

Jednakże spontaniczne, dokonywane „na zapotrzebowanie", zmiany dokumentu wymagań mogą okazać się równie niebezpieczne. Jeżeli zaakceptujemy wszystkie propozycje zmian, to ich ilość i rozmiar mogą okazać się przytłaczające. Jeżeli proponowane zmiany nie mieszczą się w (określonym wcześniej) zakresie projektu, to mimo to, że zostaną uwzględnione, nie służą osiągnięciu celów systemu. Jeżeli proponowane zmiany nie będą analizowane i następnie formalnie zatwierdzane, to do wymagań, a co za tym idzie i do projektu systemu, mogą zostać wprowadzone błędy.

W celu zminimalizowania szansy wystąpienia powyższych zagrożeń należy zdefiniować i stosować procedury kontroli zmian, a mówiąc szerzej, należy wymaganiami *zarządzać*. Stosowanie takich procedur powinno zostać przewidziane w kontrakcie zawartym pomiędzy klientem a wykonawcą. Jeżeli sposób wprowadzania zmian do specyfikacji wymagań podczas realizacji projektu nie został zdefiniowany i uzgodniony (a tym samym będzie pozostawiony przypadkowi), to należy liczyć się z wystąpieniem poważnych kłopotów.

Innym, równie istotnym aspektem zarządzania wymaganiami jest zadanie utrzymywania ich w spójności z różnymi innymi dokumentami wytwarzanymi w trakcie rozwoju systemu (takimi jak dokumentacja projektowa, implementacja, podręcznik użytkownika, podręcznik pielęgnacji, dokumentacja testów itp.).

Nigdy nie jest tak dobrze, by nie mogło być lepiej

Poprawa technologii inżynierii wymagań zwiększa gwarancje, że nakłady pracy i środków poświęcone na projekt informatyczny nie „chybią celu", tzn. zostaną przeznaczone na realizację oczekiwanego systemu. Nie daje to jednak gwarancji, że system zostanie wykonany właściwie, jego jakość będzie zadawalająca oraz będzie posiadał wszystkie oczekiwane własności. Poprawa sytuacji w tym zakresie wymaga zasadniczej zmiany, szczególnie po stronie wykonawców systemów informatycznych. Typowa sytuacja wytwórców systemów informatycznych to życie „od odbioru do odbioru", w warunkach chwiejnej równowagi pomiędzy doraźnie definiowanymi zadaniami i możliwościami wykonawczymi firmy. Zadania te są w zdecydowanej większości ukierunkowane na szybkie uzyskanie efektu w postaci działającego oprogramowania. Rezultatem jest najczęściej niska jakość produktu, co w konsekwencji powoduje spiralę poprawek i modyfikacji. Prowadzi to do chaotycznego, doraźnie zarządzanego procesu, przebiegającego wysoce niestabilnie i nie poddającego się planowaniu i sterowaniu.

Duże projekty informatyczne są postrzegane jako przedsięwzięcia o wysokim stopniu ryzyka. Nieustabilizowanie technologii realizacji powoduje, że odbiorca finansuje projekt (a często są to przedsięwzięcia o wymiarze setek osobomiesięcy pracy wysoko opłacanych specjalistów) bez możliwości wglądu w pośrednie wyniki pracy. Problemy stają się widoczne dopiero wtedy, gdy sytuacja wymyka się spod kontroli. Trzeba wtedy ponieść wysokie koszty, gdyż środki zostały już wydane, a naprawa błędów staje się bardzo kosztowna. Obniżenie ryzyka wymaga zmian zarówno po stronie odbiorców, jak i wykonawców. Odbiorcy muszą, we własnym interesie, w większym stopniu włączyć się w finansowane przez siebie projekty informatyczne. Dotyczy to przede wszystkim etapu identyfikacji projektu i formułowania wymagań. Powinni oni również dokonywać selekcji wykonawców na bazie oceny ich ***dojrzałości technologicznej***. Poprawa technologii oraz osiągnięcie odpowiedniego poziomu dojrzałości technologicznej powinno stać się jednym z celów strategicznych wykonawców projektów informatycznych. Z jednej strony polepszy to ich pozycję negocjacyjną w kontaktach z klientami, z drugiej zaś strony zwiększy skuteczność planowania oraz efektywność wykorzystania własnych zasobów.

A więc weźmy się do roboty!

Naszkicowany powyżej kierunek zmian wskazuje również nowe obszary rynku zaawansowanych technologii informatycznych. Liczący się odbiorcy i producenci zastosowań informatyki będą zainteresowani pomocą w formułowaniu i ocenie zawieranych kontraktów oraz poprawą technologii realizacji i oceny tworzonych systemów informatycznych. Szybka poprawa w tym zakresie jest możliwa jedynie przez skorzystanie z pomocy niezależnych instytucji wyspecjalizowanych w transferze technologii. Podaż tego typu usług na rynku krajowym jest jak dotąd znikoma. Stworzenie pełnej i kompetentnej oferty w tym zakresie jest poważnym wyzwaniem dla krajowego środowiska informatycznego. Szybkość, z jaką taka oferta zostanie skompletowana i udostępniona na krajowym rynku zastosowań informatyki, może mieć zasadnicze znaczenia dla skuteczności i konkurencyjności naszej gospodarki.

Janusz Górski, Katedra Zastosowań Informatyki,
Politechnika Gdańska
ul. Narutowicza 11/12, 80-952 Gdańsk
e-mail: jango@pg.gda.pl

Zasady tworzenia systemów informatycznych

Grażyna Hołodnik-Janczura

Stworzenie sprawnych i niezawodnych systemów informatycznych wymaga właściwej metodyki ich realizacji i sprawnych procesów zarządzania. Procesy te były dotychczas rzadko opisywane, a jeszcze rzadziej, przynajmniej u nas, stosowane. Przyczyny tego stanu rzeczy są dość złożone, a jedną z nich jest trudność opracowania realnego planu realizacji takiego przedsięwzięcia.

W celu zmiany tej sytuacji, korzystając ze sprawdzonych wzorów w procesach produkcyjnych, mimo istniejących trudności z konieczną standaryzacją, zaadaptowano osiągnięcia z tej dziedziny do zarządzania pracami informatycznymi. Wykorzystywane specjalistyczne narzędzia programowe z jednej strony wspomagają pracę menedżera projektu, a z drugiej stanowią okazję do gromadzenia wiedzy z tego zakresu w postaci rzetelnej bazy danych dla prowadzenia badań nad optymalizacją tego procesu.

Celem przedsięwzięcia informatycznego jest otrzymanie gotowego produktu, tzn. działającego systemu informatycznego, spełniającego oczekiwania zleceniodawcy, ale też i wykonanie go na czas, w ramach ustalonych środków finansowych. Niestety, często dzieje się tak, że zaplanowany na początku przedsięwzięcia budżet jest znacznie przekraczany, a zaplanowane terminy realizacji nie są dotrzymywane. Przyczyny są dość złożone i niełatwo jest opracować realny plan dla takiego przedsięwzięcia.

Przez przedsięwzięcie informatyczne rozumie się proces tworzenia systemu informatycznego, prowadzonego przy zastosowaniu odpowiedniej metodyki[1], technik i narzędzi, który rozpoczyna się od pierwszych badań organizacji i trwa do czasu udostępnienia gotowego systemu do eksploatacji, włączając także do tego procesu również jego rozwój. Obejmuje nie tylko proces informatyzacji, ale i komputeryzacji, które powinny być ze sobą powiązane i podporządkowane potrzebom danego przedsiębiorstwa. O wielkości tego przedsięwzięcia decydują zazwyczaj prowadzone badania wstępne, odnoszące się do wymagań wobec przyszłego systemu informatycznego. Wówczas też dokonuje się przybliżonej oceny najważniejszych zasobów: czasowych i finansowych, które są uściślane dopiero po przeprowadzeniu szczegółowych badań. Zarówno w poziomie wiarygodności tej oceny, jak i w sposobie zarządzania realizacją zaplanowanego przedsięwzięcia, należy dopatrywać się warunków jego powodzenia.

Analogicznie do innych przedsięwzięć produkcyjnych, prowadzących do powstania produktów finalnych, takich jak np. urządzenie czy budowla, proces tworzenia systemu informatycznego zaliczamy również do procesów inżynierskich. Produkty tych obu procesów, od pomysłu do wykonania, przechodzą kolejne etapy, wobec których używa się nieco odmiennej terminologii. Przy tym upodobnieniu nie można jednak pominąć istniejących różnic. W wypadku procesów produkcyjnych funkcjonują i obowiązują normy dotyczące ich wykonywania, zarówno w sferze realizacyjnej, jak i organizacyjnej, co pozwala na ścisłe kontrolowanie i dokładne mierzenie rezultatów zaplanowanych działań. Natomiast w wypadku procesów informatycznych podobna procedura normalizacyjna jest trudna do zastosowania, przede wszystkim ze względu na problemy związane z mierzeniem planowanych i realizowanych produktów, a także często na indywidualny a nie seryjny charakter tego procesu.

Zarządzanie przedsięwzięciem informatycznym

Proces zarządzania w pełni zorganizowanym przedsięwzięciem informatycznym wymaga realizacji odpowiednich funkcji zarządzania. Ze względu jednak na jego specyfikę różne będą, w porównaniu do innych przedsięwzięć inżynierskich, wymagania co do sposobów tego zarządzania, gdyż:

- *System informatyczny jest produktem niematerialnym*, efekt widoczny jest na samym końcu. Podczas tworzenia menedżer dokonuje analiz postępu prac na podstawie raportów wykonywanych przez pracowników;
- *Tworzenie systemu informatycznego jest procesem niestandardowym* i nie ma tak długiej historii, jak procesy inżynierskie. Nie jest również tak dobrze opisany. Trudne jest przewidywanie prawdopodobnych problemów szczególnie w procesie tworzenia specyficznego oprogramowania;

[1]Tworzenie systemu informatycznego powinno być poprzedzone sprawdzeniem warunków wyboru metodyki tworzenia w celu ustalenia podejścia, jak i metod realizacji całego przedsięwzięcia.

- *Nowo utworzone systemy muszą działać razem z wcześniejszymi*, często wykonanymi w przestarzałej technologii;
- *Konieczność ciągłego szkolenia i zbierania doświadczeń*. Szybkie zmiany w technologii komputerowej i komunikacji wychodzą często poza wcześniejsze doświadczenia i nie mogą być wówczas wykorzystane w nowych przedsięwzięciach;
- *Potrzeba tworzenia systemów elastycznych*. Częste zmiany zachodzące w funkcjonowaniu przedsiębiorstwa wymagają możliwości ich uwzględniania zarówno w trakcie tworzenia systemu, jak i jego eksploatacji;
- *Konieczność współpracy pomiędzy pracownikami przedsiębiorstwa a twórcami systemu*. Ograniczona możliwość tej współpracy czy brak zainteresowania prowadzonymi pracami analitycznymi ze strony użytkownika końcowego prowadzi do ustalenia rzeczywistych wymagań dopiero na etapie wdrożenia lub eksploatacji systemu.

Wymienione cechy podkreślają znaczenie indywidualnego podejścia do procesu tworzenia systemu, ale tym bardziej świadczą o potrzebie standaryzacji wszystkich powtarzalnych jego elementów. Korzystając ze wspomnianej wyżej analogii do procesów produkcyjnych, należy pracę menedżera przedsięwzięcia informatycznego wspomagać narzędziami programowymi zarówno na etapie budowy planu przedsięwzięcia, jak i w trakcie jego realizacji. Muszą to być narzędzia, które pozwolą na szybką analizę efektów wprowadzanych pomysłów rozwiązań oraz umożliwią monitorowanie i stosowanie odpowiednich mechanizmów kontroli. Dodatkową korzyść stanowić wówczas będą informacje o wszystkich zdarzeniach zachodzących w trakcie jego realizacji, zarejestrowane w bazie danych, możliwe do wykorzystania w planowaniu następnych podobnych przedsięwzięć.

Na specjalistycznym rynku oprogramowania są oferowane profesjonalne pakiety narzędziowe do zarządzania przedsięwzięciem informatycznym, zintegrowane z narzędziami do komputerowego wspomagania tworzenia systemów. Zdefiniowany wówczas opis realizacji całego przedsięwzięcia jest nazywany projektem. W tym celu mogą być również wykorzystane programy do zarządzania przedsięwzięciem produkcyjnym, ale wtedy niezbędne jest korzystanie z interfejsów do przenoszenia danych pomiędzy środowiskiem zarządzania a środowiskiem tworzenia.

Zarządzanie przedsięwzięciem informatycznym, niezależnie od stosowanych narzędzi, wymaga rzetelnego zaplanowania całego przedsięwzięcia, kontroli postępów prowadzonych prac i reagowania w wypadku wystąpienia problemów

Ramy organizacyjne procesu tworzenia systemu informatycznego zapewnia metodyka. Bez względu na przyjęty sposób podejścia, strukturalny czy obiektowy, w każdym wypadku określa ona preferowany model cyklu życia systemu, definiuje podział na etapy, definiuje standardowe procesy i produkty, a także procedury kontroli jakości i kontroli postępu prowadzonych prac. Wszystkie te elementy powinny być wykorzystane również w celu efektywnego zarządzania.

Ryzyko w realizacji przedsięwzięcia jest związane przede wszystkim z narzuconymi ograniczeniami, które dotyczą zasobów przydzielonych do dyspozycji jego wykonawcom. Zwykle jest tak, że zaplanowane wielkości zasobów czasowych i finansowych, bez dodatkowych renegocjacji, nie mogą być przekroczone. Natomiast przesunięcia pomiędzy wewnętrznymi elementami planu muszą być poprzedzone symulacją przeprowadzanych modyfikacji i analizą efektów, czy aby nie spowodują naruszenia ustalonych parametrów globalnych.

Najważniejsze miejsca decyzyjne w przedsięwzięciu informatycznym wyznaczają terminy zakończenia zadań, w wyniku których są otrzymywane tzw. produkty kluczowe, dostarczane klientowi i przez niego weryfikowane. Specyfikacje tych produktów[2] udostępnia metodyka tworzenia systemu informatycznego. Wyniki analizy poziomu wykonania zdefiniowanych produktów oraz zużycia zaplanowanych zasobów będą stanowić podstawę podejmowanych decyzji, dotyczących dalszej realizacji kolejnych punktów planu.

Decyzje na poziomie ogólnym mogą być podejmowane na podstawie spostrzeżeń i rozmów, natomiast na poziomie szczegółowym powinny wynikać z przeprowadzonych analiz. Otrzymane wskaźniki ilościowe muszą dostarczyć odpowiedzi na pytania: ile wykorzystano środków oraz jaka część systemu została zrealizowana. Wówczas można przeprowadzić, np. korektę terminów, zmianę przydziału osób do zadań lub modyfikację oszacowanych pracochłonności, aby w rezultacie dostosować plan dalszej realizacji do zachodzących zmian. Potrzeba takich modyfikacji często jest skutkiem sytuacji zaistniałych w samym przedsiębiorstwie, ale i też w zespole twórców systemu.

Iteracyjny charakteru samego procesu tworzenia systemu informatycznego również narzuca konieczność modyfikacji planu w trakcie realizacji jego etapów. Każdorazowo, po wykonaniu kolejnych coraz bardziej szczegółowych badań, dotyczących wymagań odnośnie do przyszłego systemu, następuje przybliżenie planu do rzeczywistości, a odzwierciedlenie tych szczegółów prowadzi do powstania następnej wersji planu.

Zadania menedżera przedsięwzięcia informatycznego warunkujące efektywność zarządzania

Do obowiązków menedżera należy sformułowanie takiego planu, który stanowić będzie ramy dyscyplinujące przedsięwzięcie, ale nie może to być plan statyczny. Musi on podlegać modyfikacjom w taki sposób, w jaki wymaga tego postępująca realizacja projektu. Powinien obejmować elementy związane z efektywnym zarządzaniem, wynikające z ustalonej dla takiego

[2]Na przykład w metodyce CASE*Method [1] firmy Oracle do produktów kluczowych etapu analizy między innymi należą: szczegółowy diagram obiekt-związek, szczegółowy diagram hierarchii funkcji, diagram przepływu danych, definicja kryteriów akceptacji systemu przez użytkownika.

planu struktury [4]. Jednakże w szczegółach plan konkretnego przedsięwzięcia będzie zależny od samej dziedziny przedmiotowej systemu informatycznego, a także organizacji realizującej to zadanie. W skład proponowanej struktury wchodzą następujące części:

1. *Wprowadzenie*: opis celów przedsięwzięcia, zestaw ograniczeń (budżet, czas), które mają wpływ na zarządzanie;
2. *Organizacja przedsięwzięcia*: sposób zorganizowania zespołu realizatorów; uczestnicy zespołu i ich role;
3. *Analiza ryzyka*: opis sytuacji objętych ryzykiem, prawdopodobieństwo wystąpienia ryzyka, zaproponowane strategie dla obniżenia poziomu ryzyka;
4. *Wymagania związane z zasobami sprzętowymi i programowymi*: opis sprzętu komputerowego i wspomagającego oprogramowania do wykonania systemu informatycznego. Jeżeli ma być zakupiony odpowiedni sprzęt, oszacowanie jego ceny i plan zakupów;
5. *Organizacja pracy*: opis podziału przedsięwzięcia na działania, identyfikacja „kamieni milowych" i produktów związanych z każdym działaniem;
6. *Harmonogram przedsięwzięcia*: opis zależności pomiędzy czynnościami, oszacowanie czasu potrzebnego na dojście do „kamieni milowych", przydzielenie ludzi do tych czynności;
7. *Mechanizmy śledzenia i raportowania*: opis wykonania raportów dotyczących prac zarządzania, wyznaczenie terminów ich wykonania, opis sposobów śledzenia realizacji przedsięwzięcia.

Od efektywnego zarządzania przedsięwzięciem oczekuje się zapewnienia realizacji jego celów przy ustalonych założeniach. Do tego nie wystarczy prawidłowo zaplanowane przedsięwzięcie, ale konieczna jest także identyfikacja i analiza potencjalnych źródeł ryzyka. Menedżer musi uczestniczyć w pojawiających się problemach i przygotowywać pomysły ich rozwiązywania. Stosowane narzędzie programowe ma zapewnić badanie możliwości realizacji tych pomysłów, gdyż sterowanie rozdziałem zasobów przez kontrolę postępu prac wykonywanych w trakcie realizacji przedsięwzięcia informatycznego jest jednym z podstawowych warunków zmniejszenia poziomu ryzyka związanego z dotrzymaniem terminu jego realizacji.

Śledzenie postępu prac będzie efektywne, jeżeli, oprócz listy podstawowych produktów i terminów ich wykonania, zostanie opracowana dodatkowa lista wyników pośrednich. Mogą to być elementy składowe produktów, a najlepiej raporty, wykonywane przez pracowników, które pozwolą na wcześniejsze wykrywanie ewentualnych problemów. Mechanizm ten ma na celu wspomaganie podejmowania decyzji o zmianach w planie.

Z wyżej wymienionych części planu zostanie omówiony proces definiowania harmonogramu jako element, który podlega najczęstszym zmianom i stanowi zarazem podstawowy instrument menedżera w zarządzaniu przedsięwzięciem.

Budowa harmonogramu

Szczególnie ważnym zadaniem dla menedżera jest opracowanie harmonogramu przedsięwzięcia. Chociaż przedsięwzięcia mogą wydawać się podobne, to jednak różnią się metodyką realizacji, stosowanymi narzędziami i technikami. Dlatego też wymagają indywidualnego podejścia do jego opracowania. Fakt ten nie wyklucza wykorzystywania proponowanych przez programy wzorców harmonogramów, ale zwraca uwagę na konieczność każdorazowego ich dostosowywania do rzeczywistych warunków.

Kroki prowadzące do powstania definicji harmonogramu:

1. *Wyznaczenie kalendarza dni i godzin roboczych;*
2. *Sformułowanie listy produktów;*
3. *Budowa hierarchicznej listy zadań;*
4. *Ustalenie listy zasobów;*
5. *Wyznaczenie czasu trwania dla każdego zadania;*
6. *Przydział zasobów do zadań;*
7. *Ustalenie zależności pomiędzy zadaniami typu następnik-poprzednik;*
8. *Zdefiniowanie kamieni milowych;*
9. *Ustalenie wielkości ograniczeń na podstawie ścieżki krytycznej.*

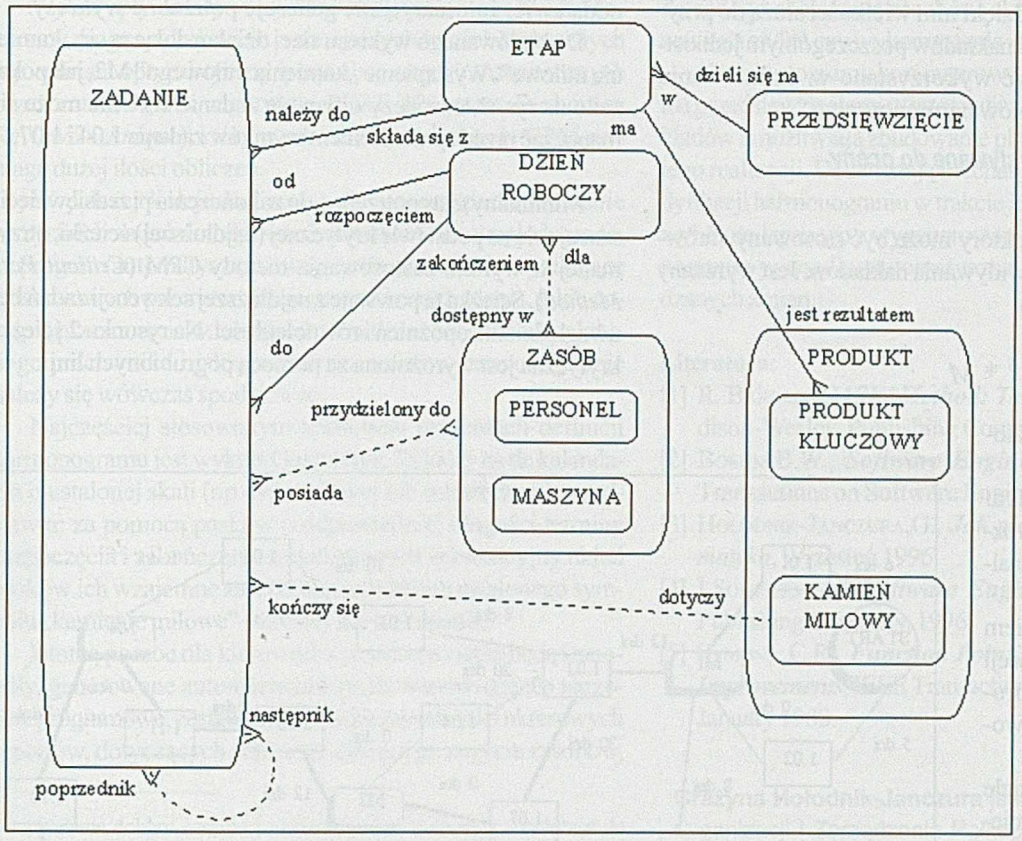
Harmonogramowanie dotyczy wydzielenia z ogółu prac oddzielnych zadań, wymaga oszacowania czasu i innych zasobów potrzebnych do ich zakończenia. Wiele zadań jest realizowanych równolegle, zatem potrzebują odpowiedniej koordynacji i takiej organizacji pracy, by dysponowany potencjał siły roboczej był wykorzystany optymalnie.

Wykonanie każdego zadania musi być dokładnie umieszczone w czasie kalendarzowym. W tym celu niezbędna jest znajomość ustawowych godzin pracy oraz dni roboczych w roku. W zależności od rodzaju przedsięwzięcia może, np. zachodzić potrzeba zaplanowania czasu na badania prowadzone z wysoko wyspecjalizowanymi urządzeniami. Wówczas będą one także ujęte przy definiowaniu i przydzielaniu zasobów.

Kierownik projektu powinien uwzględnić w harmonogramie sytuacje wyjątkowe oraz specyficzne wymagania. Nie może zakładać, że każdy etap będzie pozbawiony problemów. Pracownik może zachorować lub zrezygnować z pracy, sprzęt może ulec awarii, mogą wystąpić opóźnienia w zakupie sprzętu czy oprogramowania. Szczególnie, jeśli temat jest nowy i o dużych wymaganiach, to niektóre jego części mogą okazać się trudniejsze niż to założono i wówczas potrzeba więcej czasu niż zaplanowano wcześniej.

Dekompozycja przedsięwzięcia na etapy i zadania

Struktura podziału przedsięwzięcia zależy od przyjętej metodyki jego realizacji oraz od, uznanego za wystarczający, poziomu szczegółowości. Najczęściej jest to podział hierarchiczny, dokonywany za pomocą dekompozycji przedsięwzięcia najpierw na etapy, następnie na kroki i zadania, czasami nawet na podzadania. Zadania przeważnie nie są niezależne, zatem wymaga się przedstawienia ich wzajemnych zależności w postaci odpowiednich sekwencji.



Rys.1. Model harmonogramu w notacji diagramu Obiekt - Związek

Specyfikacja produktów

Kierowanie i kontrola przebiegu prac nie będzie możliwa bez dostarczania materialnych i mierzalnych produktów jako rezultatów wykonywanych zadań w trakcie trwania przedsięwzięcia. Do nich należą przede wszystkim raporty, ale mogą być to także wyniki demonstrowane za pomocą innych środków. Produkty powstają na końcu każdego z etapów, takich jak analiza czy projektowanie. Są niezbędne dla powodzenia przedsięwzięcia i muszą być dostarczane klientowi do jego akceptacji.

Kamienie milowe

Otrzymanie specyficznych produktów, tzw. „kamieni milowych" (*milestone*), jest wyznaczane w terminie zakończenia pojedynczego zadania lub grupy zadań, szczególnie wtedy, gdy kończy się jakieś istotne dla procesu działanie. W tym czasie menedżerowi przedsięwzięcia powinien być zaprezentowany raport o postępie prac. Inne mogą być wyznaczone, np. na koniec każdego wyodrębnionego etapu. Raporty, które pełnią rolę „kamieni milowych", w odróżnieniu od produktów, nie są dużymi dokumentami. Mają to być zwięzłe sprawozdania o osiągnięciach w prowadzonych działaniach.

Nie ma konieczności zakończenia każdego zadania „kamieniem milowym". Jeśli występują za często, to zespół cały swój czas poświęca na ich przygotowanie. Jeśli jednak są zbyt rzadko, to problemy mogą zostać nie zauważone.

Możliwość bezpośredniego definiowania „kamieni milowych" w przedsięwzięciu jest jednym z powodów szerokiego stosowania „kaskady" jako modelu cyklu życia. Charakteryzuje się ona kilkustopniową strukturą tego procesu oraz wymaganiem otrzymywania na każdym z poziomów wielu różnego rodzaju dokumentów, które raportują wykonywane czynności. Nie jest to model odpowiedni dla technologii obiektowego tworzenia systemów informatycznych, ale z powodzeniem może być stosowany do podejścia strukturalnego.

Oszacowanie pracochłonności

Istotną komplikacją przy układaniu harmonogramu jest oszacowanie wielkości (długości) jego podstawowych składników - zadań decydujących o podziale zasobów. Wielkość ta reprezentuje czas trwania każdego zadania. Zarówno czas, jak i liczba osób przeznaczonych do wykonania każdego zadania, wiążą się z wyznaczeniem pracochłonności tych zadań. Ze względu na istniejące trudności w obliczaniu pracochłonności zadań związanych z pracami informatycznymi, należy odwołać się do stosowanych, w tym wypadku, metod estymacji.

Trudności te są powodowane przede wszystkim brakiem powszechnie stosowanych norm, takich, jakie istnieją dla inżynierskich procesów produkcyjnych, które określałyby, np. ile godzin potrzeba do wykonania danej jednostki programu. Występują tutaj rozbieżności zarówno w ustaleniu jednostki miary programu, jak i w przyjęciu podstawy do przewidywania jego wielkości na wczesnym etapie tworzenia. Dodatkowym aspektem tych problemów jest także potrzeba indywidualnego podejścia do realizacji systemu informatycznego, jak to już wcześniej podkreślono, z kłopotliwymi do ustalenia nakładami prac badawczych.

Aktualnie stosowane[3] metody estymacji umożliwiają ustalenie niezbędnego poziomu nakładów pracy w odpowiednich jednostkach czasowych dla oszacowanego rozmiaru programu. Najczęściej jednostką pomiaru wielkości programu są albo punkty funkcyjne [5], albo linie kodu źró-

[3]Najczęściej opisywane w literaturze zastosowania metod dotyczących szacowania wielkości potrzebnych nakładów [3] prezentują modele, które zakładają, że związek wymaganego nakładu tworzenia oprogramowania z jego rozmiarem jest wyrażony poprzez odpowiednie równanie, którego parametry są dobierane empirycznie.

dłowego [2]. Otrzymane dzięki nim wielkości służą do przydzielenia wyliczonej ilości nakładów poszczególnym jednostkom projektu i mogą być wykorzystane w definiowaniu i modyfikowaniu szczegółów harmonogramu.

Modele i zasady, wykorzystywane do oceny pracochłonności zadań

- ***Model algorytmiczny***, który może być stosowany zarówno do oceny, jak i przewidywania nakładów. Jest wyrażany w ogólnej postaci jako:

$$Naklad = C * PM^{S} * M .$$

- C jest czynnikiem poziomu złożoności;
- PM jest miarą produktu i występuje jako miara rozmiaru lub miara funkcjonalności;
- M jest współczynnikiem otrzymanym z kombinacji atrybutów procesu, produktu i środowiska tworzenia;
- S jest zazwyczaj niewiele większe od 1, odzwierciedla wzrost nakładu dla dużego projektu;

- ***Ocena przez analogię.*** Jest wykorzystywana wtedy, gdy istnieje skompletowana dokumentacja innych projektów z tej samej dziedziny aplikacji;
- ***Zasada Parkinsona.*** Wielkość nakładów jest określana raczej przez dostępne zasoby niż przez oszacowaną wielkość produktu.

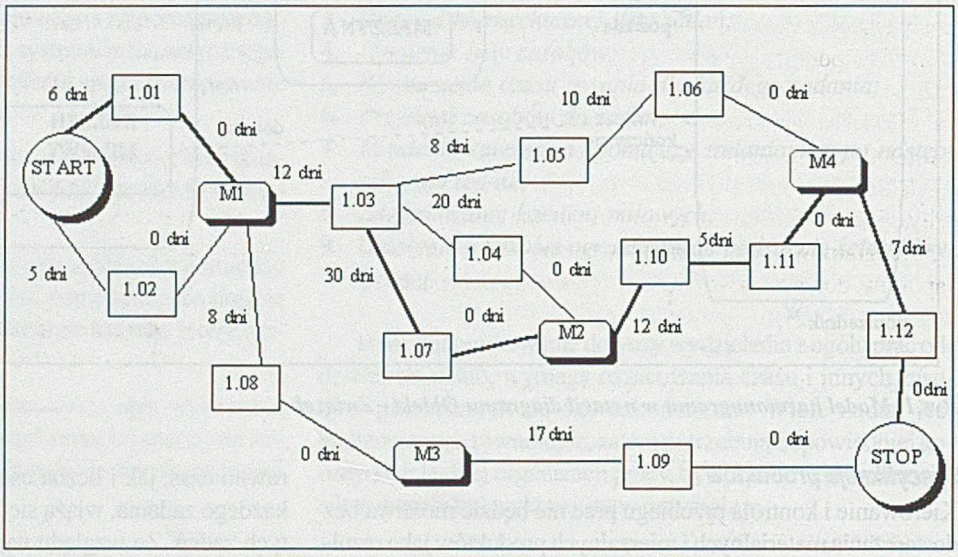
W praktyce, ze względu na „słabe" i „mocne" strony każdej z technik, stosowane są ich różne kombinacje.

Sieć działań i ścieżka krytyczna

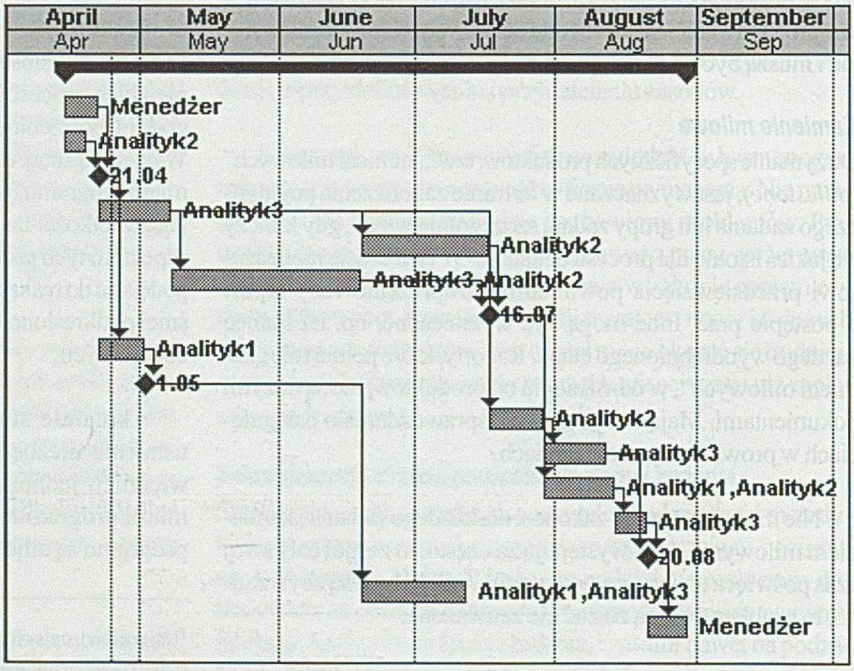
Harmonogramy najczęściej są prezentowane w postaci zbioru wykresów, ukazujących podział przedsięwzięcia, zależności pomiędzy jednostkami podziału oraz przydziały zasobów do tych jednostek. Należą do nich notacje graficzne wykresu sieci działań oraz wykresu paskowego. Sieć działań przedstawia zależności pomiędzy różnymi zadaniami. Wykres paskowy prezentuje czas rozpoczęcia i zakończenia każdego zadania oraz kto za każde z nich jest odpowiedzialny. Narzędzie programowe umożliwia zapamiętanie tych informacji w bazie danych oraz, na ich podstawie, automatycznie generuje potrzebne wykresy.

Do budowanego wykresu sieci działań dołącza się „kamienie milowe". Wystąpienie „kamienia milowego" M2, jak pokazano na rys. 2, świadczy o tym, że zadanie 1.10 nie może się rozpocząć przed sprawdzeniem wyników zadania 1.04 i 1.07.

Minimalny czas potrzebny do zakończenia przedsięwzięcia szacuje się na podstawie krytycznej (najdłuższej) ścieżki, otrzymanej w wyniku zastosowania metody CPM (*Critical Path Method*). Ścieżka ta powstaje z najdłuższej sekwencji zadań bez uwzględniania opóźnień i równoległości. Na rysunku 2 ścieżka krytyczna jest wyróżniona za pomocą pogrubionych linii.



Rys. 2. Sieć działań



Rys. 3. Harmonogram w postaci wykresu Gantta

Wybór optymalnej wersji harmonogramu może być też dokonany drogą analizy wielu potencjalnych ścieżek krytycznych (np. wariant optymistyczny, pesymistyczny), konstruowanych przez zastosowanie algorytmu PERT (*Program Evaluation Review Technique*), który w odróżnieniu od metody CPM wymaga dużej ilości obliczeń.

W dalszej kolejności rejestruje się przyporządkowanie wszystkich zadań do odpowiednich osób[4]. Ze względu na to, że program automatycznie uwzględnia opóźnienia wynikające z ustawowych przerw w pracy, prowadzi się badania, jakie nastąpią przesunięcia w terminach w razie nieobecności konkretnego pracownika i jakiego rodzaju zmian w harmonogramie należy się wówczas spodziewać.

Najczęściej stosowanym sposobem prezentacji definicji harmonogramu jest wykres Gantta (rys. 3), który na tle kalendarza o ustalonej skali (np. tygodniowej lub miesięcznej) przedstawia: za pomocą pasków o odpowiedniej długości terminy rozpoczęcia i zakończenia zadań, poprzez sekwencyjny układ pasków ich wzajemne zależności, za pomocą ustalonego symbolu „kamienie milowe" i terminy ich wykonania.

Istotną pomoc dla kierownika przedsięwzięcia będą stanowiły, generowane automatycznie przez wspomagające narzędzie programowe, zestawy zarówno bieżących, jak i okresowych raportów, dotyczących analiz np. dziennego zużycia zasobów, dziennego obciążenia zasobów, aktualnego poziomu zaawansowania zadań czy wykorzystania zasobów w zadanym czasie.

Zatem, programy komputerowe do wspomagania zarządzania przedsięwzięciami w integracji z metodami do estymacji nakładów umożliwiają zbudowanie planu o zmniejszonym ryzyku jego realizacji. Dostarczają mechanizmów kontrolowanej modyfikacji harmonogramu w trakcie jego realizacji i pozwalają na wybór najlepszego z wygenerowanych wariantów tego harmonogramu w wyniku automatycznej analizy efektów wprowadzanych zmian.

[4]Zagadnienie doboru zespołu i kierowanie jego współpracą, należące do rozważań psychologicznych i socjologicznych nie będzie tutaj omawiane.

Literatura:

[1] R. Barker, ***CASE*Method: Tasks and Deliverables***, Addison-Wesley Publishing Company, 1990.

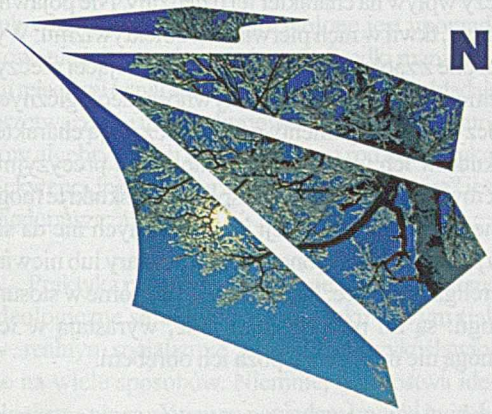
[2] Boehm,B.W., ***Software Engineering Economics***, IEEE Transactions on Software Engineering, January 1984.

[3] Hołodnik-Janczura,G., ***Jak płacić programistom***, *Informatyka*, Wrzesień 1996.

[4] I.Sommerville, ***Software Engineering***, Addison-Wesley Publishing Company, 1996.

[5] Symons, C.R., ***Function Point Analysis: Difficulties and Improvements***, IEEE Transactions on Software Engineering, January 1988.

Grażyna Hołodnik-Janczura jest pracownikiem Instytutu Organizacji i Zarządzania Politechniki Wrocławskiej



Nasze rozwiązania działają, ponieważ słuchamy naszych Klientów.

Internet
- Konta e-mail
- Dostęp dial-up
- International ROAM

Internet dla Firm
- Konta grupowe
- Dedykowane łącza
- Bezpieczne sieci korporacyjne
- Systemy bezpieczeństwa i kontroli dostępu

Usługi WWW dla Firm
- Serwery WWW
- Projekty prezentacji
- Bieżąca opieka
- Dynamiczne katalogi, ankiety, sklepy
- Specjalizowane aplikacje dla rozwiązań "business to business"

Intranet
- Pełna integracja obiegu informacji w firmie z Internetem
- Systemy "Groupware"
- Systemy "Workflow"
- Systemy śledzenia zadań i projektów
- Składnice danych
- Dedykowane systemy baz danych
- Aplikacje Lotus Notes (Domino), Oracle, Progress, Informix, Microsoft BackOffice

INTERNET - EXTRANET - INTRANET - INTERNET - EXTRANET - INTRANET - INTERNET - EXTRANET - INTRANET - INTERNET

Polska OnLine
INTERNET SYSTEM INTEGRATOR
ul.Emaliowa 28, 02-295 Warszawa

tel. (48-22) 868 0808
fax (48-22) 846 6271
http://www.pol.pl
e-mail : info@pol.pl

Obiektowość w bazach danych – koncepcje, nadzieje i fakty (3)

Obiektowość kontra model relacyjny

Kazimierz Subieta

Pojawienie się obiektowych baz danych jako konkurencji dla baz relacyjnych zburzyło spokój zarówno dostawców systemów, jak i ich klientów. Dla dostawców rynek baz danych, szacowany na 5-7 mld.$ rocznie, może być źródłem zarówno ogromnego sukcesu finansowego, jak i klęski. Z punktu widzenia klientów inwestycje w bazy danych są kosztowne, zaś dzisiejsze decyzje mogą mieć konsekwencje w przeciągu kilkudziesięciu lat.

Gorąca debata dotycząca tego, jakie systemy będą lepsze w bliższej i dalszej przyszłości – relacyjne, obiektowe czy też obiektowo-relacyjne – ma charakter sporu ideologicznego, w którym argumenty dotyczące użytkowych i technicznych własności oferowanych systemów są przemieszane z argumentami ideologicznymi i „naukowymi". W każdej z tych płaszczyzn uczestnicy debaty generują zarówno argumenty racjonalne, jak i mnogość mitów, demagogii i przekrętów.

Niewiele osób obserwujących komputerową scenę z pewnego dystansu zdaje sobie sprawę, że „nauki komputerowe" (*computer science*) nie są „nauką" w tradycyjnym rozumieniu tego terminu. Rozwój w dziedzinach związanych z komputerami oznacza przede wszystkim wynalazczość i nowe zastosowania. Działalność ściśle naukowa, zmierzająca do wypracowania obiektywnych zasad, teorii, metodologii i klasyfikacji, jest nieskuteczna, zwykle spóźniona w stosunku do aktualnych potrzeb i zbytnio obciążona zastanymi paradygmatami i kryteriami naukowymi. Przykładem mogą służyć matematyczne teorie w dziedzinie inżynierii oprogramowania, które od lat dryfują w kierunku scholastycznych mielizn, pozostawiając po sobie wrakowisko pseudowiedzy, pseudopodstaw i pseudoautorytetów. Wynalazczość nie podlega kryteriom naukowym. Jest spontanicznym błyskiem intelektu wynikającym z wiedzy, doświadczenia, wyczucia potrzeby rynkowej, wyczucia szansy zarobienia pieniędzy. Bill Gates, uważany za jednego z największych wynalazców komputerowych naszych czasów, człowiek, który w ogromnym stopniu przesądził o charakterze współczesnej informatyki, nie ukończył nawet studiów.

Model relacyjny i obiektowość można również uważać za wynalazki, chociaż o szczególnym charakterze. Są one ideologiami, czyli wynalazkami określającymi generalne ramy, cele i kierunki rozwoju. Ideologiczny aspekt oprogramowania można porównać ze stylem w budownictwie lub z ideologiami społecznymi. Ideologia wyznacza ogólne założenia co do celów, zasad i ograniczeń projektowanych systemów, określa oczekiwania użytkowników i potencjalne własności użytkowe. Nawet jeżeli system został zbudowany spontanicznie, w reakcji na kolejne potrzeby (co w informatyce często się zdarza), okazuje się, że pod powierzchnią inżynierskich działań tkwiła jakaś ideologia, chociaż nie nazwana i nieuświadomiona. Koncepcje hierarchicznych i sieciowych baz danych były właśnie tego rodzaju nieuświadomionymi ideologiami, które zostały nazwane kilka lat po ich zaistnieniu w rzeczywistości informatycznej.

W bazach danych ideologie, zwane modelami danych, okazały zasadniczy wpływ na charakter tej dziedziny. Nie pojawiły się przypadkowo, tkwił w nich pierwiastek obiektywizmu. Wyrosły bowiem one z zaobserwowanych wad istniejącej rzeczywistości. Niemniej arbitralność, subiektywność ideologicznych ustaleń jest oczywista. Argumenty na ich rzecz mają charakter werbalny. Skutków ich wprowadzenia nie da się precyzyjnie zmierzyć, jak również (patrz ideologie społeczne) skutki te mogą być sprzeczne z obietnicami. Racji ideologicznych nie da się udowodnić *a priori*: stanowią one przedmiot wiary lub niewiary, są bliższe religii niż nauce. Nauka, teorie są wtórne w stosunku do ideologii: są im podporządkowane, wyrastają w ich granicach i mogą nie mieć sensu poza ich obrębem.

W wypadku modelu relacyjnego ideologia została sklejona z teorią, ponieważ zasadniczym ideologicznym założeniem było oparcie koncepcji baz danych na matematycznej teorii relacji. Brak oddzielenia ideologii od teorii wielu zmylił. Od końca lat 70. do dzisiaj pokutuje stereotyp, zgodnie z którym model relacyjny jest uniwersalną teorią baz danych, która zmieniła charakter tej dziedziny z empirycznej na naukową. Nic bardziej błędnego: koncepcje obiektowe pokazały naocznie, że ci, którzy nie wierzą w relacyjną ideologię, mogą ją w całości wyrzucić do śmieci łącznie ze wszystkimi „naukowymi" ustaleniami, które w jej ramach zostały poczynione. (Tak jak ideologię socjalizmu, tak przecież „naukową" w opinii niegdysiejszych jej wyznawców i propagatorów).

Jakkolwiek relacyjne bazy danych stały się komercyjnym sukcesem, ich filozofia bardzo odbiegła od początkowych ide-

ologicznych ustaleń. Te wątpliwości doprowadziły E.Codda do sformułowania słynnych 12 reguł „prawdziwego" systemu relacyjnego, zaś C.Date'a do ostrych napaści na SQL jako źródło ideologicznych wypaczeń. Utyskiwania i gromy purystycznych ideologów na niewiele się zdały. Rzeczywistość relacyjnych baz danych ustabilizowała się w stanie, który jest kombinacją założeń ideologicznych, potrzeb użytkowników, obecnych możliwości technologii komputerowych, szans zarobienia pieniędzy. „Prawdziwego" systemu relacyjnego prawdopodobnie już nigdy nie będzie.

Jak wynika z tych obserwacji, porównanie relacyjnych i obiektowych baz danych wymaga wyróżnienia warstwy ideologii, która wyznacza generalne założenia dla przyszłych działań, warstwy teorii, które wyrosły lub mają szanse wyrosnąć w obrębie danej ideologii, oraz warstwy technologii, która powstała lub powstanie w efekcie realizacji przyjętej ideologii.

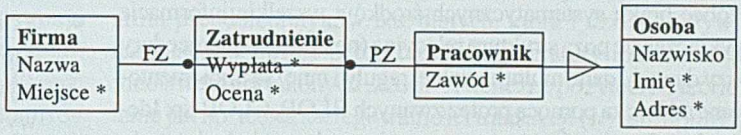
Warstwa ideologii

Podstawową doktryną modelu relacyjnego jest traktowanie bazy danych jako zestawu matematycznych relacji. Niedozwolone jest używanie wewnętrznych identyfikatorów krotek (wierszy relacji) i wskaźników prowadzących do krotek. Niedozwolone jest, aby element krotki był złożony z dwóch lub więcej wartości. Niedozwolone jest uporządkowanie lub powtórzenia krotek. Środki manipulacji relacjami (czyli środki budowy aplikacji) zakładają przetwarzanie makroskopowe za pomocą operatorów algebry relacji lub innego tego rodzaju języka. Sekwencyjne przetwarzanie „krotka po krotce" jest niedopuszczalne.

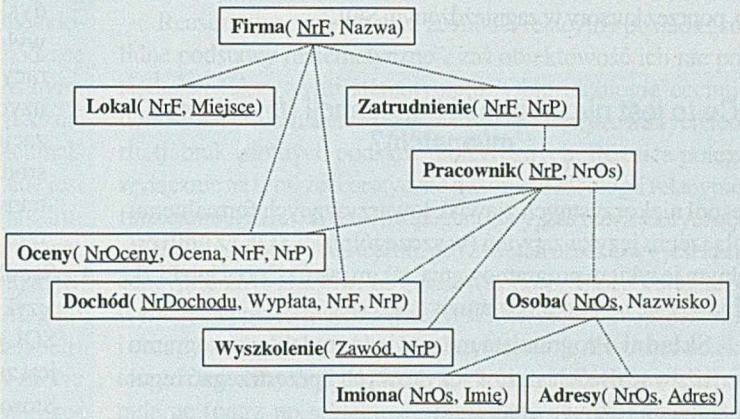
Praktyka pokazała, że opisane wyżej założenia ideologiczne są utopijne, w związku z czym (jak w „realnym socjalizmie") zostały one wykoślawione na wiele sposobów. Niemniej odstępstwa ideologiczne nie są głównym powodem krytyki modelu relacyjnego. Można nawet zgodzić się, że ustabilizował sie on w rozsądnym stanie, który wprawdzie nie pasuje do teorii relacji, ale pasuje do popularnych wyobrażeń o organizacji baz danych. Główną wadą modelu relacyjnego jest to, co miało być jego zaletą, mianowicie prostota struktur danych. W modelu relacyjnym informacje o pojęciach wyróżnialnych w rzeczywistości modelowanej przez dane są rozproszone w krotkach wielu tablic, co burzy związek pomiędzy pojęciowym obrazem świata a pojęciowym obrazem struktur danych modelujących ten świat. Skojarzenie tych informacji musi nastąpić explicite w programach aplikacyjnych (np. w zapytaniach SQL), przez co wzrasta ich koncepcyjna złożoność.

Co to jest niezgodność pomiędzy schematem pojęciowym i relacyjnym?

Poniżej znajduje się obiektowy schemat pojęciowy w notacji OMT. Klasa Pracownik jest specjalizacją klasy Osoba. Pracownik może pracować w wielu firmach, firmy są powiązane z pracownikami przez obiekty Zatrudnienie. Gwiazdką zaznaczyliśmy atrybuty powtarzalne, np. pracownik może mieć wiele zawodów. Taki schemat może być bez zmian zaimplementowany w systemach obiektowych opartych na standardzie ODMG i może być bezpośrednio użyty do formułowania zapytań w OQL i pisania programów np. w C++.



Co się stanie po przekształceniu tego schematu na postać wymaganą przez system relacyjny? Zamiast 4 klas, 2 związków i jednej generalizacji otrzymamy 10 schematów relacyjnych i 11 koncepcyjnych (nieformalnych) związków. Atrybuty powtarzalne muszą być zaimplementowane jako nowe tablice, konieczne są nowe atrybuty zawierające klucze. Informacja o pracowniku Kowalskim, która na poprzednim schemacie była skupiona w jednym obiekcie, została rozproszona w krotkach pięciu tablic.



Zapytanie „Podaj adresy pracowników pracujących w firmach zlokalizowanych w Radomiu":

OQL: **select** p.Adres **from** Firma **as** f, f.FZ.PZ **as** p **where** "Radom" **in** f.Miejsce

SQL: **select** a.Adres
from Lokal **as** k, Zatrudnienie **as** z, Pracownik **as** p, Osoba **as** s, Adresy **as** a
where k.Miejsce = „Radom" **and** k.NrF = z.NrF **and** z.NrP = p.NrP
and p.NrOs = s.NrOs **and** s.NrOs =a.NrOs

Zapytanie w SQL jest znacznie dłuższe od zapytania w OQL głównie wskutek tego, że pojawiły się w nim „złączeniowe" predykaty (takie jak k.NrF=z.NrF i następne), które kojarzą informację semantyczną zgubioną podczas odwzorowania schematu obiektowego na schemat relacyjny. To skojarzenie, oprócz wymienionej porcji dodatkowego kodu, wymaga od programisty dokładnego rozumienia nieformalnej semantyki schematu.

Brak środków hermetyzacji i modularyzacji oraz nieoddzielenie implementacji od specyfikacji są wykroczeniami przeciwko zasadom abstrakcji i dekompozycji. Są podstawą pojęciowego modelowania programów i danych. Brak złożonych obiektów, ograniczenie konstruktorów typów danych do jednego (relacji), brak wyspecjalizowanych środków do realizacji związków asocjacyjnych pomiędzy danymi, brak klas i dziedziczenia, brak możliwości ortogonalnej kombinacji różnych cech modelu są poważnymi ograniczeniami mechanizmów modelowania pojęciowego i potencjału ponownego użycia.

Model relacyjny nie zajmuje się informacją proceduralną, która dość często jest istotną składową pojęciowego obrazu danych. Wobec braku systematycznych środków wszelkie informacje wykraczające poza strukturę relacyjną (perspektywy, procedury bazy danych, dane multimedialne, reguły i inne) są implementowane ad hoc za pomocą protez zwanych BLOB, CLOB itp. Ideologia relacyjna jest oparta na naiwnych poglądach co do środków przetwarzania informacji. Trzonem tych środków miały być operatory algebry relacji lub rachunku relacyjnego; niestety, okazały się one zbyt mało uniwersalne dla prawie wszystkich zastosowań. Duża część tego przetwarzania musiała być oddelegowana do uniwersalnych języków programowania, co doprowadziło do niekorzystnego efektu niezgodności impedancji (*impedance mismatch*). W konsekwencji postulowane przez model relacyjny przetwarzanie makroskopowe wróciło częściowo do krytykowanego niegdyś przetwarzania niskiego poziomu, np. poprzez kursory w zagnieżdżonym SQL.

Co to jest niezgodność impedancji (*impedance mismatch*)?

Zespół niekorzystnych zjawisk towarzyszących formalnemu połączeniu języka zapytań (w szczególności SQL) z uniwersalnym językiem programowania, takim jak C, Cobol lub PL/I. Objawia się niezgodnościami w zakresie:

- **Składni**. Programista musi w jednym tekście programu używać dwóch stylów językowych i przestrzegać reguł dwóch różnych gramatyk;
- **Systemu typów**. Język zapytań operuje na typach zdefiniowanych w schemacie bazy danych, m.in. relacjach, natomiast język programowania posiada zwykle odmienny system typów, w którym nie występuje typ relacja. Większość języków programowania ma wbudowaną statyczną kontrolę typów, podczas gdy SQL takiej kontroli nie przewiduje;
- **Semantyki i paradygmatów języków**. Koncepcja semantyki języków jest zasadniczo różna. Język zapytań bazuje na stylu deklaracyjnym (*co* wyszukać a nie *jak*), podczas gdy języki programowania bazują na stylu imperatywnym (*jak* zrobić a nie *co*);.
- **Pragmatyki użycia**. Język zapytań uwalnia programistę od wielu szczegółów organizacji i implementacji danych (np. organizacji zbiorów, obecności lub nieobecności indeksów itd.), podczas gdy w języku programowania te szczegóły muszą być oprogramowane explicite;
- **Faz i mechanizmów wiązania**. Języki zapytań są oparte na późnych wiązaniach (są interpretowane), podczas gdy języki programowania zakładają wczesne wiązanie (podczas kompilacji i konsolidacji). Stwarza to problemy m.in. dla programów odpluskwiających;
- **Przestrzeni nazw i reguł zakresu**. Język zapytań i język programowania posiadają własne przestrzenie nazw, które mogą zawierać identyczne nazwy o różnych znaczeniach. Odwzorowania pomiędzy przestrzeniami nazw wymagają dodatkowych środków syntaktycznych i semantycznych. Przestrzeń nazw języka programowania jest zbudowana hierarchicznie i podlega regułom zakresu opartym na zasadzie stosu. Te reguły są ignorowane przez język zapytań, powodując wiele trudności;
- **Traktowania wartości zerowych**. Bazy danych i języki zapytań posiadają wyspecjalizowane środki dla przechowywania i przetwarzania wartości zerowych. Środki te nie występują w językach programowania;
- **Schematów iteracyjnych**. W języku zapytań iteracje są wtopione w semantykę operatorów, takich jak selekcja, projekcja i złączenie. W języku programowania iteracje muszą być organizowane explicite za pomocą pętli *for*, *while*, *repeat* lub innych. Przetwarzanie wyników zapytań za pomocą języka programowania wymaga specjalnych udogodnień, takich jak kursory i iteratory;
- **Traktowania cechy trwałości danych**. Języki zapytań przetwarzają wyłącznie trwałe dane (znajdujące się na dysku), podczas gdy języki zapytań przetwarzają wyłącznie dane nietrwałe znajdujące się w pamięci operacyjnej. Połączenie języków wymaga od programisty użycia specjalnych środków językowych do parametryzacji zapytań przez zmienne języka programowania oraz środków językowych i architektonicznych służących do transmisji danych z dysku do pamięci operacyjnej i odwrotnie;
- **Środków programowania ogólnego** (*generic*). Środki te w języku zapytań są oparte na refleksji (patrz np. dyna-miczny SQL). Użycie podobnego środka w języku programowania jest zazwyczaj niemożliwe z powodu wczesnego wiązania. Stosowane są inne środki, takie jak funkcje wyższego rzędu, *casting*, przejście na niższy poziom językowy, polimorfizm lub szablony.

Co zmieniła obiektowość? Przede wszystkim odrzuciła sztuczne ograniczenia związane z matematycznym rodowodem ideologii relacyjnej. Podstawowym postulatem jest możliwość tworzenia, przechowywania i przetwarzania dowolnie złożonych obiektów, które mają bezpośrednie konotacje z obiektami dziedziny przedmiotowej budowanego systemu. Informacja o pracowniku Kowalskim jest skupiona w jednym obiekcie, niezależnie od tego, jak ona jest obszerna. Może tam być seria zdjęć Kowalskiego i cała historia jego zatrudnienia. Obiekty są wyposażone w wewnętrzny identyfikator, co umożliwia tworzenie bezpośrednich powiązań wskaźnikowych pomiędzy obiektami. Nawigacja wzdłuż tych wskaźników znacznie upraszcza programowanie; sprzyja też wydajności. Wszelkie informacje wspólne dla pewnego zestawu podobnych obiektów są gromadzone w ramach klas. W szczególności dotyczy to typu obiektów oraz operacji, które na tych obiektach można wyko-

nać. Obiekty są dostępne wyłącznie przez interfejs, co umożliwia ukrycie informacji zbędnej na danym etapie programowania, na tej samej zasadzie, jak w koncepcji modułów. Przesłanianie, przeciążanie, późne wiązanie i polimorfizm są cechami sprzyjającymi czytelności, modyfikowalności i ponownemu użyciu oprogramowania. Systemy obiektowe zakładają kompletność obliczeniową i pragmatyczną środków programistycznych, co eliminuje niezgodność impedancji.

Podsumowując, model relacyjny przegrał konfrontację z obiektowością w warstwie ideologicznej. Poczynając od ok.1985 roku, nastąpiła coraz intensywniejsza krytyka, w wyniku której świat naukowy zredukował swoje zainteresowanie tym modelem praktycznie do zera. Dzisiaj nie jest widoczny ośrodek, który aktywnie rozwijałby model relacyjny od strony ideologiczno-naukowej. Obiektowość jest natomiast przedmiotem bardzo intensywnego rozwoju w setkach ośrodków. Wbrew poglądom głoszonym przez zwolenników łączenia ideologii relacyjnej z obiektową, model relacyjny w warstwie ideologicznej nie oferuje obiektowości niczego istotnego głównie z tego powodu, że obiektowe struktury danych zawierają struktury relacyjne jako szczególny przypadek[1].

Warstwa teorii

Podstawową wadą w zasadzie wszystkich teorii zbudowanych dla potrzeb modelu relacyjnego jest ich dekoracyjny, fasadowy charakter. Teorie te miały wprawdzie pewne znaczenie metodologiczne, próbowały wyjaśnić zjawiska semantyczne zachodzące w danych i w językach zapytań na gruncie matematycznym. W konsekwencji jednak okazały się dla praktyki bezużyteczne. W szczególności dotyczy to tzw. teorii zależności funkcjonalnych, wielowartościowych i innych, których zastosowanie jest znikome i sprowadza się do definicji kilku pojęć (takich jak druga, trzecia i czwarta forma normalna) o znaczeniu marginalnym dla praktyki projektowania baz danych. Oparcie semantyki języków zapytań takich jak SQL na algebrze relacji lub rachunku relacyjnym okazało się pomysłem nieudanym. Jak wspomnieliśmy, jednym z powodów jest to, że systemy relacyjne w istocie odchodzą od pojęcia relacji. „Relacje" przechowywane w systemach relacyjnych są strukturami bogatszymi niż relacje matematyczne, m.in. mogą zawierać powtórzenia krotek, zaś istotną rolę odgrywają w nich klucze oraz nazwy atrybutów. Ponadto, algebra relacji lub rachunek relacyjny są niedostatecznie uniwersalne dla języków takich jak SQL. Na przykład nie uwzględniają operacji aktualizacyjnych, funkcji zagregowanych, grupowania, uporządkowania, operatorów arytmetycznych i innych.

Mitem są częste twierdzenia, że metody optymalizacji zapytań są konsekwencją odkrytych własności algebry relacji. Historycznie metody optymalizacyjne (np. przesuwanie selekcji i projekcji przed złączenie) zostały zaimplementowane kilka lat przed odkryciem podobnych praw algebry relacji. Ponieważ algebra ta nie przykrywa i nie pasuje do wielu aspektów języka SQL, jakakolwiek własność odkryta w tej algebrze niekoniecznie musi obowiązywać w SQL i odwrotnie. Bardziej zaawansowane metody optymalizacyjne (np. wyznaczanie optymalnej ścieżki dostępu) są rezultatem głębokich praktycznych dociekań, heurystyk i eksperymentów i nie są podbudowane jakąkolwiek istotną teorią.

Całkowitym fiaskiem zakończył się duży nurt teoretyczny określany jako „badania nad niekompletną informacją". Nie są znane jakiekolwiek próby zastosowania tych rezultatów, nawet w prototypach. Natomiast rozwiązania w tym zakresie w języku SQL są pozbawione elementarnej logiki (patrz artykuły C. Date'a). Podobnie całkowitym fiaskiem zakończyły się rozszerzenia teoretyczne modelu relacyjnego, takie jak uniwersalne relacje i dedukcyjne bazy danych (setki publikacji, dziesiątki doktoratów). Negatywnym skutkiem tych „badań" było wykształcenie armii pseudoteoretyków baz danych, która i dzisiaj wpływa hamująco na postęp przez zamulanie literatury i konferencji mnóstwem poronionych „teorii". Nieliczne pożyteczne teorie, takie jak np. teoria szeregowalności transakcji (*serializability*), są w gruncie rzeczy neutralne w stosunku do modelu danych i równie dobrze nadają się do modeli obiektowych.

W systemach „postrelacyjnych" lub „obiektowo-relacyjnych" termin „relacyjny" ma znaczenie wyłącznie tradycyjno-dekoracyjne. Systemy te, poprzez rozbudowę własności struktur danych i języków przetwarzania danych, nie są w stanie skorzystać z własności matematycznego pojęcia relacji w duchu relacyjnego modelu danych. Jest dość wątpliwe, czy dla tych pomysłów (w istocie generowanych przez praktyków) ktokolwiek będzie budować teorie.

Reasumując, twierdzenia, że model relacyjny posiada „solidne podstawy matematyczne", zaś obiektowość ich nie posiada, są fałszywymi stereotypami. Obie ideologie cechuje dokładnie ten sam status względem zmatematyzowanych teorii, tj. brak istotnych podstaw teoretycznych. Różnica polega wyłącznie na tym, że teoretyczne dekoracje ideologii relacyjnej funkcjonują jako czynnik propagandowy, zaś teorie dotyczące obiektowości nie pełnią tej roli, gdyż model obiektowy jest koncepcyjnie zbyt złożony, aby pasował do prostoty i elegancji wymaganej przez matematykę. Wskutek zbytnich uproszczeń i obciążenia pomysłami z epoki relacyjnej teorie dotyczące obiektowych baz danych są jak dotąd wadliwe, nieskuteczne i niepopularne (patrz np. obiektowe algebry lub logiki, koncepcje „dedukcyjno-obiektowe" itd.). Tworzą one jednak już tło, z którego może wyłonić się autentyczna, niedekoracyjna teoria obiektowych baz danych. Przykładem może służyć intensywnie rozwijana teoria typów polimorficznych w zastosowaniu do obiektowości.

Warstwa technologii

Sprzymierzeńcem wszystkich wdrożonych technologii baz danych, w tym relacyjnych, jest ogromna bezwładność rynku zastosowań, który niechętnie zmienia swoje preferencje, szczególnie jeżeli zostały zainwestowane duże pieniądze i czas. W dziedzinie baz danych obowiązuje model nisz ekologicznych. Jeżeli jakaś koncepcja technologiczna opanowała pewną dziedzinę zastosowań (np. bankowość) i/lub teren geograficzny, wówczas bardzo trudno zastąpić ją inną koncepcją, nawet jeżeli jest ona obiektywnie lepsza. Klient baz danych nie tylko nie lubi kosztownych zmian. Musi mieć także pewność, że nie po-

[1] Wskutek tego każdy system obiektowy jest jednocześnie systemem obiektowo-relacyjnym; ten galimatias komercyjnej retoryki powoduje mnóstwo nieporozumień.

zostanie sam w swojej dziedzinie działalności lub rejonie geograficznym i może liczyć na zarówno środowisko specjalistów, jak i ogólną kulturę techniczną wytworzoną w związku z daną technologią.

Systemy relacyjne opanowały dużą grupę „nisz ekologicznych" i można przyjąć jako pewnik, że pozostaną w nich przez kilka, kilkanaście lub nawet kilkadziesiąt lat. Systemy obiektowe muszą poszukiwać innych nisz, które nie są zagospodarowane przez wcześniejsze technologie. Wielu specjalistów wskazuje, że potencjalnym polem zastosowań obiektowych baz danych są multimedia, dziedziny wymagające bardziej rozbudowanych struktur danych, takie jak CAD (*Computer Aided Design*), CAM (*Computer Aided Manufacturing*), CASE, lub dziedziny implikujące nieregularne, niesformatowane struktury danych, takie jak zastosowania pełnotekstowe, hurtownie danych, zastosowania biurowe. Prace standardyzacyjne prowadzone m.in. przez OMG i ODMG wzmacniają ten rynek przez usystematyzowanie i ukierunkowanie rozwoju, tworzenie powszechnie akceptowanej terminologii, pojęć i kultury technicznej oraz przez wytworzenie wśród potencjalnych klientów poczucia dojrzałości technologicznej i niezależności od jednego dostawcy.

Prawie wszystkie systemy relacyjne przechodzą etap radykalnych modernizacji (m.in. a może przede wszystkim) w związku z wyzwaniem, jakie stawiają technologie obiektowe. Następuje wzmocnienie struktur danych przechowywanych w systemach relacyjnych o nowe możliwości. Rozszerzenia te wprowadzają wiele elementów obiektowości. Wiele systemów relacyjnych jest wyposażanych w możliwość współdziałania z innymi systemami (w tym nierelacyjnymi) według standardów obiektowych OMG CORBA i COM/DCOM. Zwolennicy modelu relacyjnego bronią tej koncepcji przez tworzenie systemów relacyjno-obiektowych. Spontaniczność i niesystematyczność tego rozwoju powoduje bardzo niską kompatybilność systemów. Ponad 95% programów rzekomo „zgodnych" ze standardem SQL-92 nie daje się przenieść na inne platformy SZBD, głównie ze względu na niepełność tych standardów oraz niekompatybilne rozszerzenia implementowane w poszczególnych systemach. Specjaliści powątpiewają co do implementowalności SQL3: ogrom tego standardu prawdopodobnie spowoduje, że będzie on implementowany w (różnej) części na różnych platformach, co może podważyć sens standardyzacji. W istocie więc następuje kryzys koncepcji w systemach relacyjnych oraz ich ewolucja w kierunku obiektowości. Trudno przewidzieć, gdzie ta ewolucja się zatrzyma.

Dość często porównanie obiektowości i relacyjności schodzi na grunt własności technicznych. Technologiczna przewaga systemów relacyjnych nad obiektowymi jest jednak prawie na pewno kwestią czasu i odpowiednich inwestycji.

Nie istnieje użytkowa własność systemów relacyjnych, która nie mogłaby być równie dobrze zrealizowana w systemach obiektowych

Nie widać istotnego argumentu na korzyść tezy przeciwnej. Np. zwolennicy systemów relacyjnych zarzucają systemom obiektowym, że nie są one wyposażone w odpowiednie funkcje, np. zintegrowane środowiska do tworzenia aplikacji, ale są to argumenty doraźne, ponieważ być może za miesiąc lub rok te funkcje będą już obecne. Wiele zarzutów technicznych pod adresem systemów obiektowych ma charakter dowolnych twierdzeń powtarzanych przez tendencyjną konkurencję lub ludzi niezbyt kompetentnych w przedmiocie.

Koronnym zarzutem informatycznych wszechczasów jest wydajność (*performance*), np. kiedyś była ona argumentem zwolenników CODASYL-u przeciwko modelowi relacyjnemu, później (po wynalezieniu automatycznej optymalizacji zapytań) stała się argumentem odwrotnym. Dzisiaj również argument ten jest odbijany jak piłeczka ping-pongowa. Zwolennicy relacyjności zarzucają systemom obiektowym brak dopracowanych metod optymalizacji zapytań, zaś zwolennicy obiektowości podkreślają zredukowanie potrzeby tej optymalizacji po zaimplementowaniu związków asocjacyjnych jako wskaźników. Tymczasem obiektowość sprzyja wydajności również przez technikę przemiany wskaźników (*pointer swizzling*), indeksy ścieżkowe (*path indices*) oraz przez nowe mechanizmy indeksacji i buforowania umożliwiającymi istotne przesunięcie przetwarzania na stronę klienta w architekturze klient-serwer. Niektóre testy wydajnościowe wykazują, że systemy obiektowe są sprawniejsze od relacyjnych tysiące razy.

Częste mity, demagogie, fałszywe stereotypy i przekręty

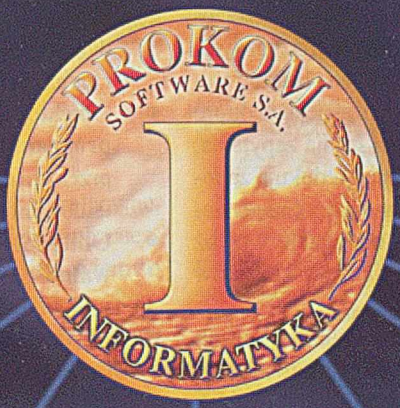
Dyskusje, opinie i zestawienia porównujące systemy relacyjne z obiektowymi zawierają bardzo dużo tez nieprawdziwych lub kontrowersyjnych. Niżej zamieszczamy komentarze na ich temat.

- **W odróżnieniu od systemów obiektowych systemy relacyjne i SQL posiadają solidne podstawy matematyczne.** Nieprawda. Struktury danych implementowane w systemach relacyjnych (tablice) odbiegają od matematycznych relacji. Jakakolwiek teoria słuszna dla relacji nie musi obowiązywać dla tych struktur. Semantyka SQL zasadniczo odbiega od ram formalnych, takich jak algebra relacji, rachunek relacyjny i logika matematyczna. SQL posiada wiele operatorów nie rozpatrywanych przez relacyjne teorie, np. operatory aktualizacyjne, uporządkowanie, grupowanie, eliminację duplikatów, operatory arytmetyczne, funkcje zagregowane itd.
- **Można skonstruować kompletny język zapytań/programowania oparty wyłącznie na algebrze relacji.** Być może, ale po co? Pomysł nadaje się wyłącznie dla teoretyków o skłonnościach masochistycznych.
- **Operatory algebry relacji spełniają tę samą rolę dla systemów baz danych jak operatory arytmetyczne dla komputera.** Demagogia. Operatory algebry relacji są daleko niewystarczające do realizacji nawet najprostszych aplikacji w systemach relacyjnych. Wewnętrzny mechanizm baz danych jest oparty na bardzo szerokim zestawie operatorów, znacznie odbiegających od operatorów algebry relacji.
- **Nie można zagnieżdżać obiektowych zapytań, ponieważ nie są one oparte na własności domkniętości** (*closure property*). Niezrozumienie tematu. Zapytania w OQL

INFORMACJA

Właściwa informacja
to dziś najbardziej pożądany towar.
Każda branża potrzebuje innych danych,
opracowanych i dostępnych w różnych formach.
Mając szybki dostęp do potrzebnych informacji,
w potrzebnej Ci formie i czasie, działasz sprawniej.
Masz przewagę nad konkurencją.

PROKOM
jest największym w Polsce dostawcą
kompleksowych rozwiązań informatycznych.
Rozwiązań, dzięki którym informacje
pracują na Twoją korzyść.



PROKOM
SOFTWARE S.A.

można zagnieżdżać pomimo (pozornego) niespełnienia własności domkniętości.

- **Systemy relacyjne umożliwiającą matematyczną weryfikację poprawności zaprojektowanej bazy danych.** Nie umożliwiają. Projektowanie relacyjnych baz danych jest z reguły oparte na całkowicie nieformalnym modelu encja-związek. Pojęcia, takie jak 2NF, 3NF, 4NF, BCNF oraz związane z nimi teorie zależności funkcjonalnych i wielowartościowych mają dla praktyki znaczenie marginalne i służą prawie wyłącznie jako pseudonaukowy muł zapychający programy nauczania i podręczniki dla studentów. Projektant instynktownie unika sytuacji, które nie są zgodne z 3NF, 4NF itd. Są sytuacje, kiedy projektant musi podjąć świadomą decyzję prowadzącą do niezgodności z 3NF i 4NF. Nie przewiduje ich teoria.
- **Standard SQL-92 zapewnia pełną przenaszalność oprogramowania pomiędzy różnymi systemami relacyjnych baz danych.** Nie zapewnia. Są opinie, że ponad 95% programów zgodnych z SQL-92 nie jest przenaszalna. Istnieje kilka powodów: (1) Semantyka SQL-92 jest wyspecyfikowana nieprecyzyjnie i pozostawia wiele luk, np. w zakresie dostępu do katalogów; (2) SQL musi być zanurzony w język programowania (np. C), którego standaryzacja jest zwykle wątpliwa; (3) Dostawcy systemów nie traktują poważnie kwestii zgodności ze standardem, czyniąc dość dowolne odstępstwa i rozszerzenia; (4) Wiele oprogramowania bazuje na mutacjach SQL (np. ODBC, JDBC, PL/SQL systemu Oracle lub języki 4GL).
- **Systemy relacyjne umożliwiają realizację dowolnego zastosowania.** Nieprawda. Dla wielu zastosowań struktury relacyjne okazują się zbyt sztywne. Odwzorowanie struktur pojęciowych na struktury relacyjne wiąże się ze znacznym wzrostem złożoności, która może podważyć osiągalność celów projektu.
- **Obecność wskaźników w strukturach obiektowych cofa rozwój do lat 70.** Demagogia. W starych systemach sieciowych programista posługiwał się wskaźnikami fizycznymi explicite, co miało liczne wady. W systemach obiektowych wskaźniki mają charakter pojęciowych asocjacji, które poprawiają percepcję schematów, upraszczają zapytania, programy i pielęgnację oprogramowania, redukują zapotrzebowanie na pamięć i umożliwiają znaczną poprawę czasów wykonania przez bezpośrednią nawigację wzdłuż wskaźników lub przez technikę przemiany wskaźników (*pointer swizzling*).
- **Model relacyjny i systemy relacyjne w unikalny sposób sprzyjają implementacji rozproszonych baz danych.** Mit uzasadniany powierzchownymi cechami, takimi jak możliwość „łatwej" dekompozycji relacyjnej bazy danych na składowe lub dekompozycji relacji na podrelacje. Te cechy nie są w stanie uprościć zasadniczych problemów rozproszenia danych, takich jak przezroczystość rozproszenia, rozproszone transakcje, przetwarzanie zapytań w sytuacji rozproszenia, replikacje, autonomia lokalnych baz danych, osłony, mediatory, perspektywy itd. Pojawienie się standardów i pakietów, takich jak CORBA, DCOM i JavaBeans, sugeruje, że dalszy rozwój rozproszonych baz danych będzie odbywał się na podstawie technologii obiektowych.
- **Systemy obiektowe nie mogą być wyposażone w języki zapytań.** Nieprawda. Patrz np. OQL.
- **Hermetyzacja jest nieakceptowalna, bo bezsensownie ogranicza bezpośredni dostęp do danych i uniemożliwia realizację języka zapytań.** Nieporozumienie wynikające ze zbyt wąskiego rozumienia pojęcia hermetyzacji. Hermetyzacja, polegająca na oddzieleniu specyfikacji (interfejsu) od implementacji i ukryciu nieistotnych szczegółów implementacyjnych, jest podstawową zasadą nie tylko obiektowości, ale całej inżynierii oprogramowania. Hermetyzacja w duchu języków Modula-2, C++, Java i Eiffel nie stwarza jakichkolwiek trudności koncepcyjnych z językami zapytań.
- **Obiektowe języki zapytań nie posiadają sprawnych metod optymalizacyjnych.** Demagogia, z kilku powodów: (1) Podstawowym celem metod optymalizacyjnych w systemach relacyjnych jest kosztowna operacja złączenia. Ponieważ w systemach obiektowych złączenie jest najczęściej zmaterializowane w postaci wskaźników (asocjacji), zapotrzebowanie na tę operację jest znacznie zredukowane; (2) Jak wykazują testy, optymalizacja relacyjnych języków zapytań nie zawsze jest tak sprawna, jak to jest eksponowane w literaturze; (3) Nie można wskazać powodów, dla których metod sprawnych w systemach relacyjnych nie można zastosować w systemach obiektowych; (4) Ortogonalność i bardziej precyzyjna semantyka OQL (w stosunku do SQL) stwarza większy potencjał dla metod optymalizacyjnych.
- **Obiektowe bazy danych mają małą wydajność.** Twierdzenie dowolne. Systemy obiektowe są o kilkanaście lat młodsze od relacyjnych, a mimo to są często znacznie od nich sprawniejsze.
- **Obiektowość jest dobra na etapie analizy i projektowania, ale jest mało przydatna dla implementacji.** Twierdzenie to wynika z nieprzystosowania zastosowanych narzędzi implementacyjnych (np. relacyjnych baz danych, baz dokumentów typu Lotus Notes itp.) do pojęć i technik obiektowych.
- **Obiektowość jest drugorzędną cechą niektórych języków programowania.** Tak od dawna nie jest. Obiektowość przerasta cały cykl analizy, projektowania i realizacji systemów informacyjnych.
- **Obiektowe systemy baz danych nie zapewniają skalowalności.** Twierdzenie dowolne. Wiele systemów obiektowych takich jak ObjectStore, Objectivity/DB lub O2 nie posiada istotnych ograniczeń co do rozmiaru bazy danych lub liczby użytkowników.
- **Obiektowe bazy danych nie zapewniają właściwych środków przetwarzania transakcji.** Nieprawda. Wszystkie liczące się systemy obiektowe zapewniają przetwarzanie transakcji.
- **Obiektowe bazy danych sprowadzają się do dodania cechy trwałości do obiektowych języków programowania.** Nieprawda. Obiektowe bazy danych są wyposażane we wszystkie niezbędne mechanizmy, takie jak: zarządzanie pamięcią zewnętrzną, schematem, współbieżnością, odtwarzaniem i transakcjami, środki bezpieczeństwa i prywatności, przetwarzanie zapytań, interfejsy do szybkiego tworzenia aplikacji, środki wspomagające rozproszone bazy danych i inne.

Obiektowość – potencjalne ryzyko

Jak każda ideologia obiektowość nie jest pozbawiona wad lub zagrożeń. Ich listę (prawdopodobnie niekompletną) podajemy niżej.

- Niedopracowane mechanizmy zarządzania dużą bazą obiektów, sterowania wersjami, rejestrowania zmian, zapewnienia stabilności interfejsów;
- Technologie obiektowe są jak dotąd stosowane przez małe i średnie organizacje. Nie jest do końca pewne jak przeskalują się dla wielkich organizacji;
- Duża liczba tematów znajduje się ciągle w fazie laboratoryjnej;
- Szereg technologii jest mało stabilnych (np. metodyki analizy i projektowania);
- Przejście na technologie obiektowe może zagrozić funkcjonowaniu obecnie działających i sprawnych systemów, które są krytyczne dla misji organizacji;
- Zbyt mała liczba ekspertów jest wyszkolona w zakresie technologii obiektowych;
- Nie jest jasne, jakie koszty pociągnie za sobą przejście na technologie obiektowe;
- Standardy w zakresie obiektowości, takie jak: CORBA, ODMG i COM, są niedopracowane i niestabilne. Nie wiadomo, w jakim zakresie będą one pełnić swoją funkcję. Standardy te zakładają zbytni eklektyzm (wiązania do wielu języków programowania), co powoduje silne ograniczenie funkcjonalności;
- Oczekiwania związane z opanowaniem problemu rozproszonych baz danych przez obiektowe oprogramowanie komponentowe (takie jak pakiety pośredniczące wg CORBA, OpenDoc, JavaBeans lub DCOM) są przesadzone. Rozproszenie, heterogeniczność baz danych generuje wiele trudnych tematów (takich jak optymalizacja rozproszonych zapytań) znacznie wykraczających poza możliwości przewidziane przez to oprogramowanie.

❖ ❖ ❖

Czy spór dotyczący tego, czy systemy mają być relacyjne, obiektowe czy też obiektowo-relacyjne, może mieć jednoznaczne rozstrzygnięcie? Z punktu widzenia tego, co dzisiaj wiemy o systemach informacyjnych, biorąc pod uwagę ideologię, konsekwencję i estetykę produktów informatycznych, koncepcje obiektowe mają absolutną przewagę nad relacyjnymi. Świat jednak bardzo dużo zainwestował w model relacyjny, wyprodukowano dziesiątki systemów, działają tysiące zastosowań, napisano setki podręczników, wykształcono niezliczoną liczbę specjalistów. Czasu i inwestycji nie da się cofnąć i dlatego przyjdzie nam żyć z systemami relacyjnymi przez wiele następnych lat. W tym pejzażu znajdą się również obiektowe i obiektowo-relacyjne bazy danych. Nigdy nie uda się ustalić precyzyjnej granicy, gdzie kończy się „relacyjność" a zaczyna „obiektowość". Określenie „relacyjny" od dawna jest dowolnie rozciągane i zmieniane, dawno utraciło związek z matematycznym rodowodem i nie bardzo wiadomo, co dzisiaj znaczy. Pozostała z niego tylko konwencja językowa, niejasny komercyjny stereotyp, który jest coraz bardziej anachroniczny - na korzyść koncepcji i terminów obiektowych.

Kazimierz Subieta jest kierownikiem zespołu inżynierii oprogramowania w Instytucie Podstaw Informatyki PAN oraz wykładowcą w Polsko-Japońskiej Wyższej Szkole Technik Komputerowych.
e-mail: subieta@ipipan.waw.pl
WWW: http://www.ipipan.waw.pl/~subieta

VI KRAJOWA KONFERENCJA EDI-EC

Zapraszamy Państwa na kolejną VI Krajową Konferencję EDI-EC, która odbędzie się w dniach 2-4 czerwca 1998 r. w Dobieszkowie k. Łodzi. EDI (*Electronic Data Interchange*) wraz z innymi środkami elektronicznego (bez papierowego) prowadzenia procesów gospodarczych coraz częściej jest określana mianem Electronic Commerce (EC). Termin ten dobitnie charakteryzuje jakościowo nowe środowisko, w którym świat będzie realizował procesy biznesowe na przełomie XX i XXI wieku. Podobnie jak w latach ubiegłych zapraszamy do udziału w naszej imprezie przedstawicieli wszystkich polskich środowisk zainteresowanych EDI i EC. Tradycyjnie również, dla rozszerzenia płaszczyzny dyskusji oraz konfrontacji polskiej perspektywy z dokonaniami światowymi, zapraszamy na konferencję ekspertów zagranicznych. Dotychczas gościli u nas specjaliści z Australii, Austrii, Belgii, Czech, Francji, Holandii, Niemiec, Szwajcarii, Turcji, USA, Włoch, Wielkiej Brytanii oraz ONZ. Rok 1998 będzie rokiem dalszej „ekspansji geograficznej" naszej konferencji jeśli chodzi o zaproszanych gości. Przewidujemy bowiem udział m.in. eksperta z Brazylii.

PROGRAM KONFERENCJI

Kontynuując tradycje programowe naszej konferencji pragniemy zaoferować Państwu przegląd najnowszych osiągnięć w zakresie teorii i praktyki Electronic Data Interchange oraz Electronic Commerce prezentowanych zarówno przez krajowych wykładowców, jak i ekspertów zagranicznych, których wystąpienia będą tłumaczone.

Konferencja ma charakter szkoleniowy i jest adresowana do wszystkich obecnych i przyszłych twórców oraz użytkowników systemów EDI reprezentujących różnorodne dziedziny zastosowań.

Głównym celem konferencji jest korzyść poznawcza wyniesiona przez uczestników oraz dalsza konsolidacja krajowego „lobby" użytkowników systemów klasy EDI-EC.

LOKALIZACJA KONFERENCJI

Konferencja odbędzie się w hotelu „Dobieszków" oddalonym o 7 km od administracyjnej granicy miasta Łodzi. Hotel wraz z kompleksem gastronomicznym w starym stylowym dworku stanowią enklawę komfortu i nowoczesności w leśnym rezerwacie przyrody pod nazwą „Struga Dobieszkowska". Wokół hotelu znajdują się bardzo atrakcyjne tereny spacerowe. Organizatorzy zapewniają bezpłatny przejazd autokarem z łódzkich dworców kolejowych do hotelu. Na uczestników, którzy przyjadą własnymi samochodami, będzie oczekiwało bezpłatne miejsce na parkingu strzeżonym przy hotelu.

Całkowite koszty uczestnictwa jednej osoby wynoszą 690 zł przy wpłacie przed 25 marca br. lub 750 zł przy wpłacie po tym terminie. Powyższe koszty obejmują: udział w obradach, komplet materiałów konferencyjnych, 3 noclegi w pokoju 2-osobowym, pełne wyżywienie oraz program wieczorny.

Inne szczegóły dotyczące warunków uczestnictwa oraz formularz zgłoszeniowy znajdziecie Państwo w Internecie:
Email: lodz-edi-ec@post.ld.onet.pl; http://www.ld.onet.pl/lodz-edi-ec

Dodatkowych informacji udzielają: Hanna i Marian Niedźwiedzińscy „CONSULTING", ul. B. Szczodrego 22/6, 92-414 Łódź, tel/fax: (42) 670-72-86 oraz Sabina Kamińska z firmy „EDIPOL", Warszawa, tel/fax (0-22) 838 83 57.

Zagrożenia dla zamierzeń informatycznych

Marian Kuraś, Agnieszka Zając

Technika informacyjna (TI) jest techniką zarządzania (*management, organizational technology*). Jej wykorzystanie do wspomagania organizacji i zarządzania stało się czymś naturalnym. Powstają coraz bardziej złożone systemy informacyjne (SI) wykorzystujące bardziej zaawansowaną TI, co zwiększa nakłady oraz ryzyko zastosowań i trudność sterowania procesem ich tworzenia i wykorzystania.

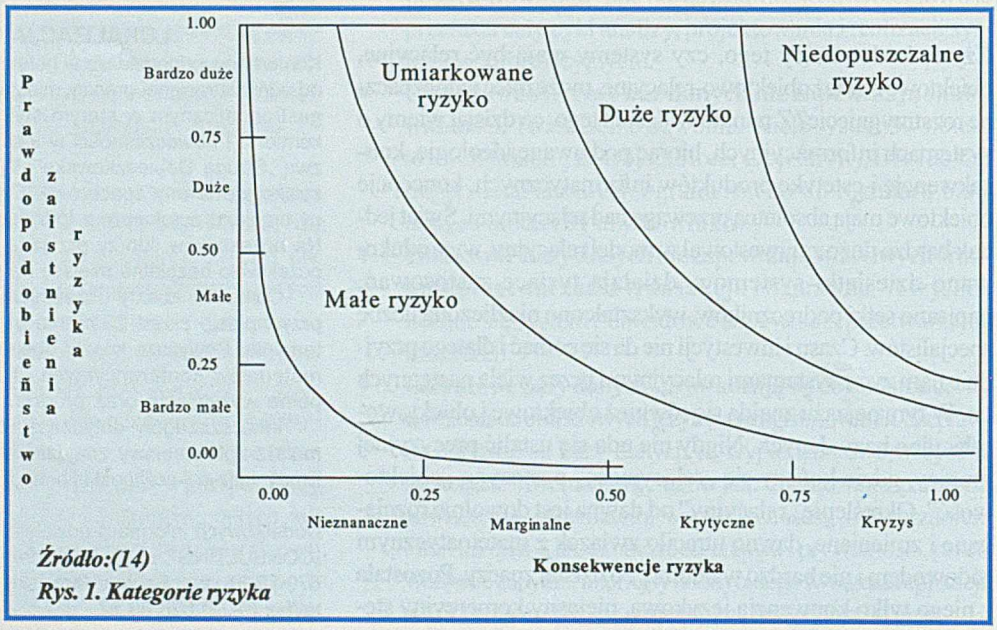
Liczne zagrożenia uniemożliwiają osiąganie celów SI. Mogą one być typowe lub specyficzne w określonych warunkach, podobnie jak i czynniki powodzenia. Zachodnia literatura z zakresu SI i zarządzania dostarcza licznych opracowań dotyczących ryzyka projektów modernizacji SI (np. 1, 2, 3, 4, 12, 14). Znacznie rzadziej spotyka się analizę projektów SI i ich niepowodzeń w naszych publikacjach (np. 5, 6, 7, 8, 13).

Istota zagrożeń i ryzyka zamierzeń

Na złożone zamierzenia oddziałują zagrożenia często powodujące ryzyko. Ich źródła są różnorodne i to może stanowić przyczynę, dla której nie podejmuje się działań mających zapobiegać ryzyku. Każdy projekt, a szczególnie kompleksowy SI, jest zamierzeniem ryzykownym, o czym przekonują się użytkownicy i twórcy SI, ale zwykle dopiero w trakcie zamierzenia lub po wdrożeniu SI. Skutki zagrożeń są przyjmowane jako dopust boży bądź cecha TI. Użytkownicy, szczególnie w początkowym okresie wprowadzania TI, nie planują jej zastosowań, nie wskazują celów projektu, a po wdrożeniu nie oceniają skuteczności i efektywności (5, 7, 8).

Po przyjęciu do eksploatacji SI nie dokonuje się zazwyczaj jego oceny. Każdy wdrożony projekt uznaje się za udany i co najwyżej stwierdza się, że zawsze można zrobić coś lepiej. 75% zamierzeń okazuje się jednak nieudane, gdyż nie odpowiadają one wcześniejszym założeniom (8, 11). Użytkownicy i twórcy SI powinni sobie zdawać sprawę z tego, co oznacza udane zamierzenie i planować je tak, aby osiągnąć sukces.

Ryzyko zamierzenia Neumann określa jako *potencjalny problem z określonymi przyczynami i skutkami oznaczającymi straty* (12). Według Sage'a (14) *ryzyko oznacza prawdopodobieństwo powstania szkód, zniszczeń lub strat w pewnym specyficznym otoczeniu, warunkach i określonym czasie.* Zagrożenia powstają i ujawniają się w fazie tworzenia SI, a następnie podczas jego wykorzystania jako np. opóźnienia, przekroczenia kosztów, zakłócenia w funkcjonowaniu, niesprawności lub niefunkcjonalności. Dopiero po zaistnieniu zakłóceń powodowanych przez nieuwzględnione wcześniej zagrożenia poszukuje się doraźnych rozwiązań lub akceptuje sytuację. Pozostają straty, których można było uniknąć. Wiele zamierzeń bowiem można przeprowadzić inaczej, identyfikując i analizując wcześniej ich ryzyko i kwalifikując je do jednej z kategorii wymienionych na rysunku 1. Efektem analizy byłoby zaniechanie projektu lub raczej zmiana planów jego wykonania z uwzględnieniem zagrożeń.



Źródło:(14)

Rys. 1. Kategorie ryzyka

Większość zagrożeń, które napotykają twórcy SI i użytkownicy, można przewidywać i przeciwdziałać im oraz ich skutkom. Jeśli nie bierze się ich pod uwagę: nie identyfikuje się ryzyka, nie analizuje, nie ocenia potencjalnych skutków i nie planuje przeciwdziałania, można być pewnym, że ujawnią się one w trakcie wykorzystania SI.

Systematyka źródeł ryzyka zamierzeń informatycznych

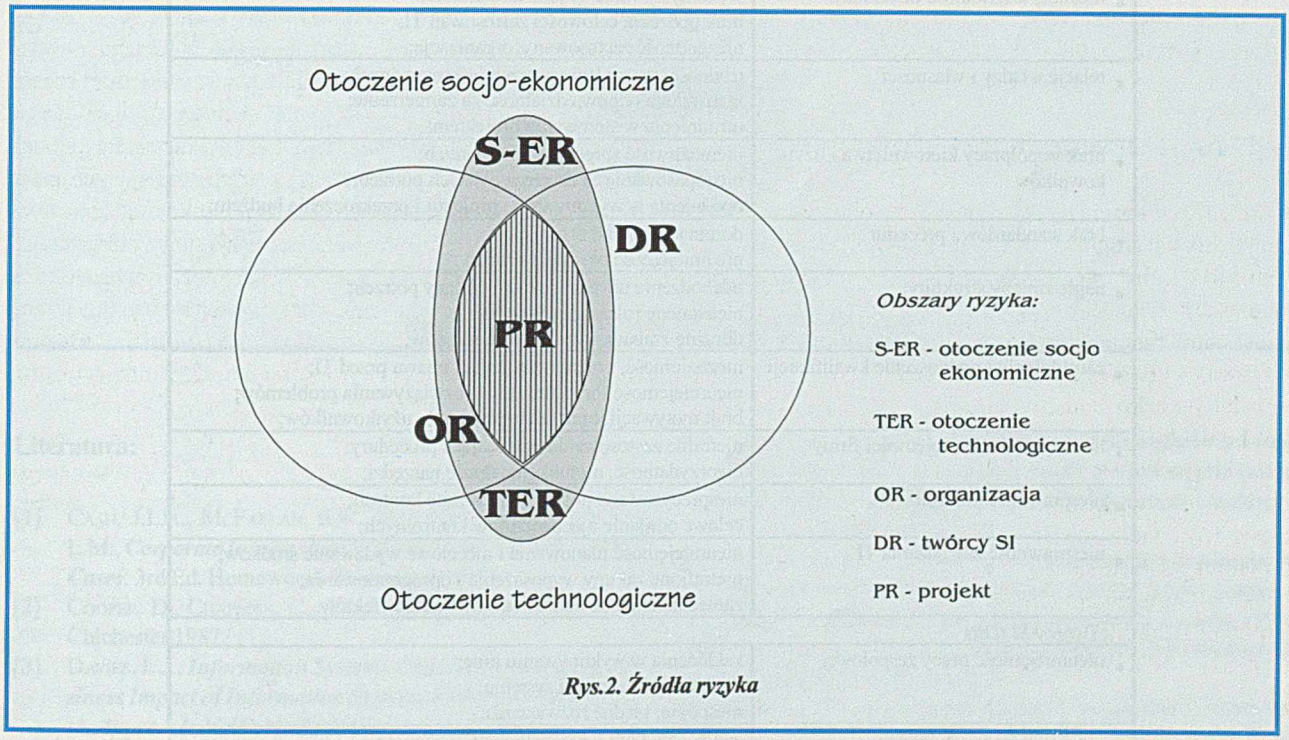
Wśród źródeł zagrożeń i wynikającego z nich ryzyka można wskazać typowe, oddziałujące na wszystkie zamierzenia podejmowane w określonych warunkach. Daje to możliwość wykorzystania doświadczeń innych użytkowników i twórców SI w celu unikania zagrożeń. Powinno się też wykorzystywać doświadczenia z wcześniejszych wdrożeń. Inne czynniki ryzyka można identyfikować, badając plan projektu i określając działania eliminujące ryzyko i/lub zmniejszające jego skutki.

Na rysunku 2 przedstawiono typologię źródeł ryzyka wynikającego z zagrożeń dla zamierzenia. Wyróżniono pięć kategorii zagrożeń, które mogą stanowić przyczynę niepowodzenia zamierzenia. Tablica 1 zestawia wybrane główne zagrożenia w każdej z wyróżnionych kategorii.

Możliwości zapobiegania i sterowania ryzykiem

Ryzyku zamierzeń można i należy zapobiegać. Minimalnym wymaganiem jest identyfikacja i analiza podstawowych zagrożeń wynikających z planu zamierzenia. Ekspertyza planu, oparta na charakterystyce projektu i organizacji, pozwala ocenić wstępne ryzyko, a następnie zmniejszać je przez modyfikacje planu. Różnorodność zagrożeń i ich złożoność wskazują na celowość tworzenia systemów sterowania kompleksowym ryzykiem zamierzeń. System taki obejmuje rozpoznawanie, analizę, planowanie i operacyjne redukowanie ryzyka.

W pracach 7, 8, 9 przedstawiono propozycję metody sterowania ryzykiem RAM (*Risk Analysis & Management*). Metoda zawiera trzy moduły (rysunek 3) umożliwiające wstępną ocenę i redukcję ryzyka (*Risk Evaluation & Analysis Procedure*), planowanie działań zapobiegawczych na trzech poziomach: proaktywnym, interaktywnym i reaktywnym (*Risk Mapping*) oraz operacyjne zapobieganie / zmniejszanie ryzyka (*Risk Abatement Performance*). Dotychczas opracowano i zastosowano procedurę REAP. Pozostałe dwa moduły będą szczegółowo opracowane, gdy pojawią się takie potrzeby użytkowników.



Rys. 2. Źródła ryzyka

Pierwotne zagrożenia wywodzą się z otoczenia, a pozostałe są ich pochodnymi. Wszystkie są wzajemnie zależne. Najbardziej znaczące jest ryzyko projektu, na które oddziałują wszystkie pozostałe. Duży wpływ mają czynniki organizacyjne, np. nastawienie użytkowników, określenie celów i strategii oraz gotowość i zdolność do przeprowadzenia zmiany organizacyjnej. Widzenie zmiany organizacyjnej jako celu wprowadzania TI jest warunkiem powodzenia zamierzenia (10).

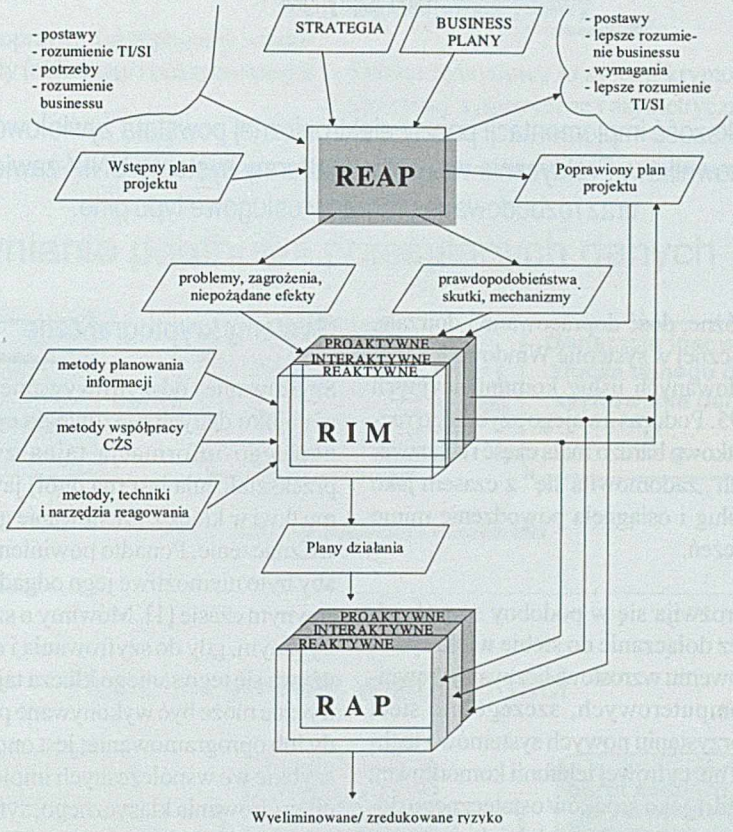
Przedstawione w skrótowej postaci zagrożenia i ryzyko zamierzeń informatycznych mają zasygnalizować potrzebę ich uwzględniania: rozpoznawania, analizy i zapobiegania/zmniejszania w planowaniu, tworzeniu i wykorzystywaniu SI. Liczne z zagrożeń można eliminować bądź zmniejszać dzięki analizie identyfikacji, analizie oraz sterowaniu kompleksowym ryzykiem. Przedstawiona systematyka zagrożeń z pewnością może być przydatna do analizy podobnie jak propozycja metody RAM.

Źródła (obszary) ryzyka	Potencjalne zagrożenia
Otoczenie społeczno ekonomiczne (SER)	
• podejście do TI	strach przed zmianą; nieumiejętność celowego zakupu i wykorzystania TI;
• niespójny/nietrwały system gospodarczy i prawny	zmiana celów i zadań organizacji; konieczność częstych zmian oprogramowania;
• zmiany sytuacji na rynku, rosnący rynek	niedostosowanie do wymogów klientów; niewydolność systemu;
• system edukacji	nieumiejętność pracy zespołowej; nieznajomość zarządzania; nieumiejętność wykorzystania narzędzi;
• brak standardów	niespójność danych; czasochłonność wprowadzania i uzyskiwania danych;
• niska kultura informacyjna	nieskuteczność zabezpieczeń; nieumiejętność wykorzystania większości funkcji TI.
Otoczenie technologiczne (TER)	
• niedorozwój telekomunikacji	opóźnienia w przesyłaniu danych i ich przekłamania; utrudniony dostęp do informacji; wysokie koszty eksploatacji;
• brak standardów przesyłania danych	pracochłonność opracowywania danych; nieczytelność danych; niewielkie możliwości wykorzystania danych;
• przewaga techniki mikrokomputerowej	nieznajomość innych technologii; niedopasowanie technologii do potrzeb; duże wydatki na TI.
Organizacja (OR)	
• nieokreślone cele oraz brak wizji i strategii	nieokreślone cele systemu informacyjnego; komputeryzowanie istniejących procedur; nieuwzględnianie potrzeb wynikających ze wzrostu firmy;
• niechęć, niezdolność do zmiany	wykorzystywanie TI jako kalkulatora; brak poczucia celowości zastosowań TI; niezgodność zastosowań z organizacją;
• relacje władzy i własności	trudności w ustaleniu potrzeb informacyjnych; nieustalona odpowiedzialność za zamierzenie; utrudnienia w sterowaniu projektem;
• brak współpracy kierownictwa i użytkowników	niemożliwość sprecyzowania potrzeb; niedopasowanie SI do rzeczywistych potrzeb; opóźnienia w wykonywaniu projektu i przekroczenie budżetu;
• brak standardów i procedur	dominacja TI nad organizacją; nieumiejętność określenia zadań ;
• nagłe zmiany struktury	odchodzenie użytkowników i zmiany potrzeb; nieustalone role organizacyjne; doraźne zmiany procedur i standardów;
• zatrudnienie i podnoszenie kwalifikacji	nieznajomość, brak zrozumienia i obawa przed TI; nieumiejętność formułowania i rozwiązywania problemów; brak motywacji, brak zainteresowania użytkowników;
• nierozpoznane umiejętności firmy	nietrafne zastosowania zakłócające procedury; nieprzydatność, niefunkcjonalność narzędzi;
• niesprawność kontroli	niesprecyzowane potrzeby dotyczące kontroli; celowe omijanie mechanizmów kontrolnych;
• niesprawność zarządzania TI	nieumiejętność planowania i niecelowe wydawanie środków; nietrafione zakupy wyposażenia i oprogramowania; zaniechanie planowania i egzekwowania efektów.
Twórcy SI (DR)	
• nieumiejętność pracy zespołowej	zakłócenia w wykonywaniu prac; błędna strukturyzacja systemu; niespójne, błędne rozwiązania;
• podejście do zamierzenia	zaniechanie lub powierzchowność analizy; pomijanie badania pracy; dostosowywanie użytkowników do TI a nie SI do ich potrzeb; 'komputeryzacja' zamiast zmiany;
• nieznajomość metod, technik i narzędzi	stosowanie metod znanych i wypróbowanych a niepotrzebnych; niekompletna analiza, niespójna specyfikacja; niewykorzystywanie możliwości narzędzi; nietrafne oceny kosztów, efektów i czasu wykonania projektu.
Projekt (PR)	
• obszerność i złożoność zadania projektowego	brak analizy problemów; trudność opanowania złożoności, nietrafna strukturyzacja;
• wycinkowe projektowanie i oprogramowywanie SI	niewłaściwa kolejność opracowania i wdrażania modułów SI; niespójność modułów systemu.

cd. tabeli

• brak business planu	nieświadomość celów oraz kosztów i efektów; nieliczenie się z kosztami, pomijanie oczekiwanych efektów.
• brak sterowania zamierzeniem	przekraczanie terminów i kosztów; zakłócenia we współpracy uczestników;

Tab. Źródła ryzyka i wybrane zagrożenia
Źródło: opracowanie własne



Rys. 3. RAM - metoda analizy i sterowania ryzykiem

Literatura:

[1] CASH, J.I.JR., MCFARLAN, F.W., MCKENNEY, J.L., APPLEGATE, L.M., ***Corporate Information Systems Management. Text and Cases***, 3rd Ed. Homewood, Ill.: Irwin. (1991).

[2] COOPER, D., CHAPMAN, C., ***Risk Analysis for Large Projects***, Chichester 1987.

[3] DAVIES, L.J., ***Information Systems Failure. Assessing the Business Impact of Information Systems Failure: A Risk Analysis Approach***, Vol.9, No.4. ,1991.

[4] HODDER, J.E., RIGGS, H.E. , ***Pitfalls in Evaluating Risky Projects***, Harvard Business Review, January-February 1985.

[5] KURAŚ, M., ***Analiza ryzyka zamierzeń informatycznych. INFOGRYF'94. Materiały konferencyjne***, tom 1, Szczecin - Kołobrzeg 1994.

[6] KURAŚ, M., ***Ryzyko wprowadzania techniki informacyjnej. Inteligentne systemy wspomagania decyzji w zarządzaniu. Mat.konf.***, Katowice 1995.

[7] KURAŚ, M., ZAJĄC, A., ***IS Project Risk at Polish Organizations. Referat przyjęty na konferencję Business Information Systems***, WI'97 w Berlinie, 1996.

[8] KURAŚ, M., ZAJĄC, A., ***Sterowanie ryzykiem zamierzeń informatycznych. Referat na konferencję: Wielodostępne systemy informatyczne do zarządzania przedsiębiorstwem***, Ciechocinek 16-18 września 1996.

[9] KURAŚ, M., ZAJĄC, A., STEFAŃSKI, M., ***Strategiczne planowanie rozwoju systemu informacyjnego. Analiza ryzyka zamierzeń informatycznych. Raport z pracy badawczej***, Kraków 1996.

[10] LEDERER, A.L,, NATH, R., ***Making Strategic Information Systems Happen***, 1990, Vol.4, No.3.

[11] LYYTINEN, K., ***Expectation Failure Concept and Systems Analysts' View of Information System Failures: Result of an Exploratory Study. Information and Management***, 1988, No.14.

[12] NEUMANN, P.G., ***Computer Related Risks***, New York 1995.

[13] POŁAP, J., ***Zarządzanie ryzykiem systemów informatycznych. Rachunkowość***, 1995, No.4.

[14] SAGE, A.P., ***Systems Management for Information Technology and Software Engineering***, N.Y.-Chichester-Brisbane-Toronto 1995.

Artykuł był prezentowany na II Konferencji „Informatyka na Wyższych Uczelniach dla Gospodarki Narodowej".

Kryptograficzna ochrona poczty komputerowej

Piotr Szpryngier

Zdecydowana większość implementacji poczty elektronicznej powstała żywiołowo jako odpowiedź na zapotrzebowanie użytkowników. Praktycznie wszystkie platformy systemu UNIX zawierają polecenie typu mail oraz rozbudowane programy usługowe typu pine.

Pojawiły się również różne, dość dopracowane i dojrzałe, programy poczty elektronicznej w systemie Windows 3.x (np. PEGASUS) aż do wbudowanych usług komunikacyjnych w systemie WINDOWS'95. Podczas swojego szybkiego rozwoju przenosiły one początkowo bardzo małą część ruchu wiadomości. Większość z nich „zadomowiła się" z czasem jako jedna z podstawowych usług i osiągnęła powodzenie mimo wielu sceptycznych zastrzeżeń.

Poczta elektroniczna rozwija się w podobny sposób jak poczta klasyczna - poprzez dołączanie do siebie wielu systemów. Rośnie dzięki lawinowemu wzrostowi liczby użytkowników wszelkich sieci komputerowych, szczególnie sieci Internet, oraz dzięki wykorzystaniu nowych systemów i technik telekomunikacyjnych (np. cyfrowej telefonii komórkowej, łączy światłowodowych itd.) jako środków ostatecznego dostarczania wiadomości. Rozwija się również dzięki towarzyszącemu jej społecznemu zainteresowaniu jako nowej możliwości komunikowania się ludzi i jako wynik wielu przedsięwzięć. Jej upowszechnieniu zaczyna towarzyszyć przekonanie, że poczta elektroniczna wywrze na społeczeństwa kulturowy wpływ równy uzyskanemu przez pocztę klasyczną lub go przewyższający [2].

Bezpieczeństwo komunikacji pocztowej, tzn. poufne i niezawodne dostarczanie przesyłek, ma kluczowe znaczenie dla pomyślnego rozwoju poczty elektronicznej. Obecnie wszystkie implementacje poczty komputerowej zapewniają jawną komunikację, tzn. przesyłane informacje mają postać kartek pocztowych. Pojawiły się jednak programy pomocnicze, które umożliwiają zamknięcie przesyłanych wiadomości w tzw. kopertach cyfrowych za pomocą nowoczesnych technik kryptograficznych.

Szyfrowanie oraz protokoły kryptograficzne zapewniają ochronę wiadomości. Umożliwiają one realizację podpisów (cyfrowych) oraz utajnianie wiadomości, naśladujące klasyczne sposoby kontroli nienaruszalności (koperty) i autorstwa (podpisy) przesyłki. Te cele osiąga się dzięki połączeniu technik kryptografii z kluczem jawnym i kryptografii symetrycznej.

Systemy kryptograficzne

Szyfrowanie i odszyfrowywanie to odwracalne przekształcenia bloku danych możliwe do odwrócenia dla kogoś dysponującego informacją tajną zwaną kluczem. Algorytm przekształcenia jest (na ogół) jawny. Bezpieczeństwo systemu tkwi w kluczu. Zachowanie go w tajemnicy ma podstawowe znaczenie. Ponadto powinien on być wystarczająco długi, aby było niemożliwe jego odgadnięcie lub wyliczenie w sensownym czasie [1]. Mówimy o szyfrowaniu (systemie) symetrycznym, gdy do szyfrowania i odszyfrowania bloku danych używa się tego samego klucza tajnego. Przekształcenie symetryczne może być wykonywane przez wyspecjalizowane układy lub oprogramowanie; jest ono stosunkowo proste i bardzo szybkie we współczesnych implementacjach. W odróżnieniu od szyfrowania klasycznego, symetrycznego, mówimy także o systemach z kluczem jawnym, tzw. asymetrycznych, w których używamy pary kluczy: klucza jawnego, powszechnie znanego, i klucza prywatnego, który powinien być utrzymywany w tajemnicy. Algorytmy z kluczem jawnym są o wiele wolniejsze (1000 razy) od symetrycznych. Ich działanie opiera się na trudnych problemach obliczeniowych, które są stosunkowo łatwe do realizacji dla posiadających dodatkową informację tajną lub klucz prywatny [1].

Najważniejszą korzyścią z istnienia algorytmów kryptograficznych z kluczem jawnym jest możliwość realizacji różnorodnych protokołów kryptograficznych (np. wymiana kluczy symetrycznych, pieniądze cyfrowe itp.) oraz dwóch podstawowych operacji zapewniających uwierzytelnienie (autorstwo) oraz poufność przesyłanych wiadomości. Ma to szczególnie znaczenie w sytuacji, gdy nadawca i odbiorca informacji nigdy się nie spotkali ani się nie znają (co jest częstą sytuacją podczas pracy w sieci). Jedyną wymaganą tu informacją jest znajomość kluczy jawnych osób korespondujących ze sobą. Te dwie operacje to podpisy cyfrowe oraz przesyłanie wiadomości zaszyfrowanej razem z zaszyfrowanym kluczem, użytym do jej zaszyfrowania. Ponieważ szyfrowanie asymetryczne jest znacznie wolniejsze, to do realizacji podpisu cyfrowego wykorzystuje się wartość funkcji jednokierunkowej, obliczo-

nej dla przesyłanej wiadomości. Skrót wiadomości nosi wiele nazw, m.in. odcisk palca (*fingerprint*), kryptograficznej sumy kontrolnej, kontroli spójności wiadomości itd. Schematy operacji zabezpieczania wiadomości zilustrowano poniżej.

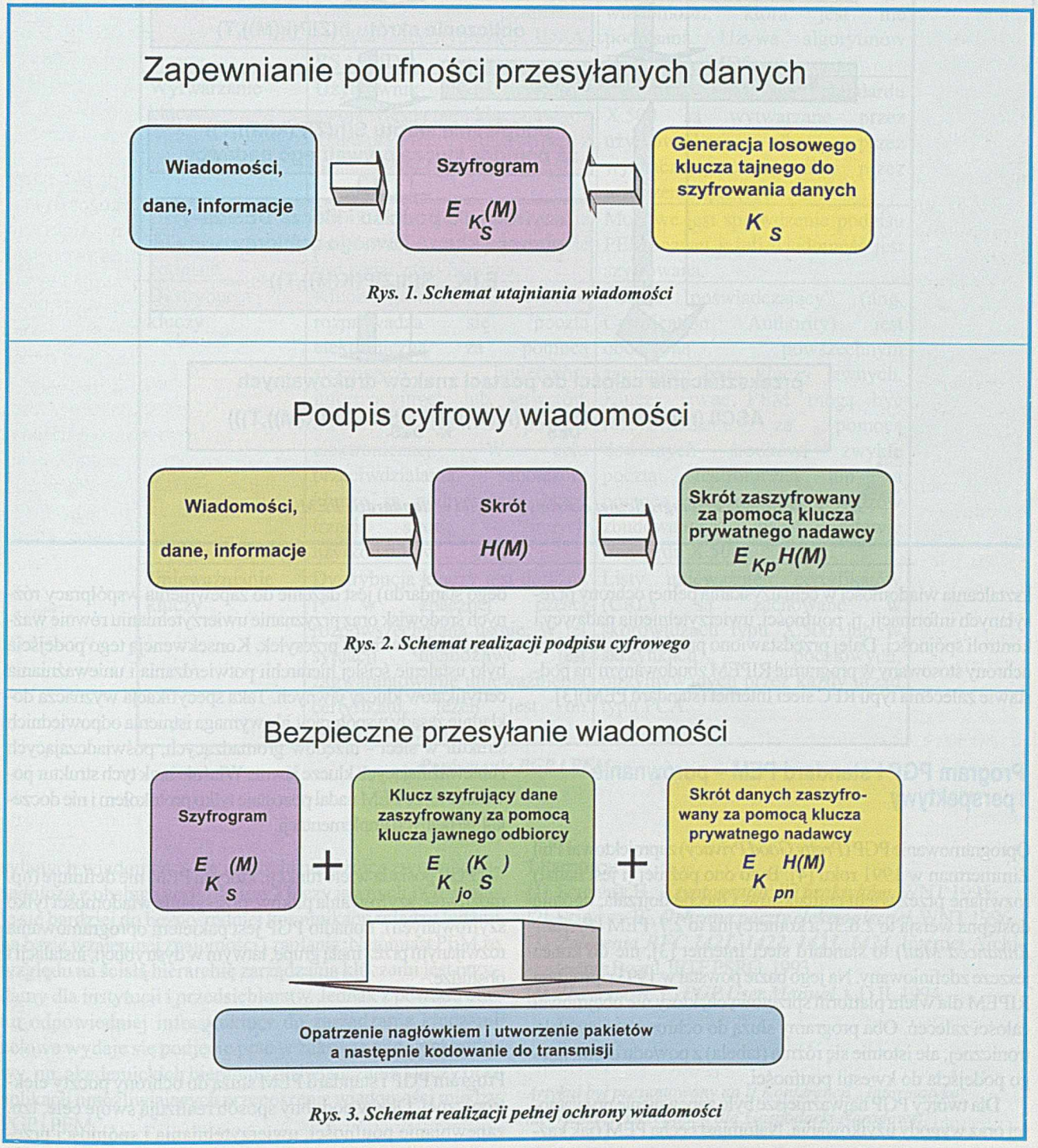
Procedura sprawdzenia podpisu wygląda następująco:
###odbieramy wiadomość i podpis
###obliczamy skrót wiadomości
###deszyfrujemy odebrany podpis za pomocą klucza jawnego nadawcy
###wynik uznajemy za poprawny (akceptujemy wiadomość) wtedy, kiedy oba skróty (obliczony i odszyfrowany z podpisu) są zgodne.

Uwaga: Ta procedura stanowi jednocześnie sprawdzenie spójności przesyłanej wiadomości. Jeżeli (jak poprzednio) wartość obliczonego skrótu wiadomości różni się od skrótu podpisanego (zaszyfrowanego kluczem prywatnym nadawcy), to można również podejrzewać, że wiadomość uległa jakimś celowym zmianom podczas transmisji.

Praktycznie łączy się te dwie operacje, aby zapewnić jednocześnie uwierzytelnienie i spójność wiadomości oraz jej poufność (rys. 3).

System hybrydowy

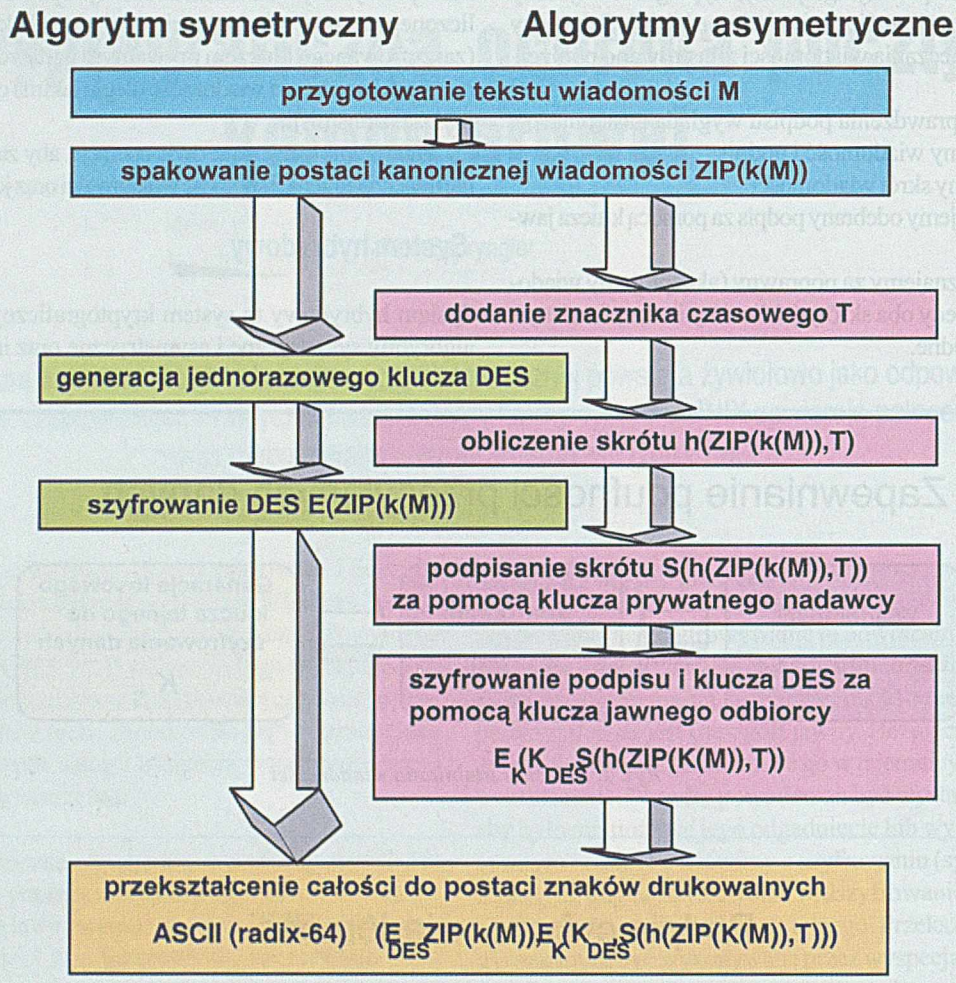
System hybrydowy to system kryptograficzny wykorzystujący algorytmy symetryczne i asymetryczne oraz inne techniki prze-



Rys. 1. Schemat utajniania wiadomości

Rys. 2. Schemat realizacji podpisu cyfrowego

Rys. 3. Schemat realizacji pełnej ochrony wiadomości



Rys. 4. Realizacja kryptograficznej ochrony przesyłki w standardzie PEM

kształcania wiadomości w celu uzyskania pełnej ochrony przesyłanych informacji, tj. poufności, uwierzytelnienia nadawcy i kontroli spójności. Dalej przedstawiono przykład realizacji tej ochrony stosowany w programie RIPEM zbudowanym na podstawie zalecenia typu RFC sieci Internet (standard PEM)[3].

Program PGP i standard PEM – porównanie i perspektywy

Oprogramowanie PGP (*Pretty Good Privacy*) zaprojektował Phil Zimmerman w 1991 roku [4]. Było ono później (i jest nadal) rozwijane przez wielu entuzjastów. Obecna dojrzała, ogólnie dostępna wersja to 2.6.3i, a komercyjna to 2.7. PEM (*Privacy Enhanced Mail*) to standard sieci Internet [3], nie do końca jeszcze zdefiniowany. Na jego bazie powstał w 1993 r. program RIPEM dla wielu platform sprzętowych, który nie obejmował całości zaleceń. Oba programy służą do ochrony poczty elektronicznej, ale istotnie się różnią (tabela) z powodu odmiennego podejścia do kwestii poufności.

Dla twórcy PGP najważniejsze było zapewnienie prywatności oraz wygody użytkowania. Natomiast cechą PEM (jak każdego standardu) jest dążenie do zapewnienia współpracy różnych środowisk oraz przyznanie uwierzytelnianiu równie ważnej roli jak poufność przesyłek. Konsekwencją tego podejścia było ustalenie ścisłej hierarchii potwierdzania i unieważniania certyfikatów kluczy jawnych. Taka specyfikacja wyznacza dokładnie zasady współpracy, ale wymaga istnienia odpowiednich struktur w sieci - urzędów gromadzących, poświadczających i unieważniających klucze jawne. Właśnie brak tych struktur powoduje to, że PEM nadal pozostaje tylko protokołem i nie doczekał się pełnych implementacji.

PGP posiada wiele funkcji, których PEM nie definiuje (np. możliwość szyfrowania plików, przesyłania wiadomości tylko szyfrowanych). Ponadto PGP jest pakietem oprogramowania rozwijanym przez małą grupę, łatwym w dystrybucji, instalacji i obsłudze.

❖ ❖ ❖

Program PGP i standard PEM służą do ochrony poczty elektronicznej. Oba w podobny sposób realizują swoje cele, tzn. zapewnianie poufności, uwierzytelniania i spójności prze-

porównywana cecha	PGP	PEM
Model zaufania	sieciowa dystrybucja zaufania	hierarchiczna dystrybucja kluczy
Zastosowania docelowe	Zaprojektowano do ochrony poczty elektronicznej, ale jest także bardzo użyteczny do zabezpieczania wybranych plików binarnych. Prywatność uważa się za co najmniej równie ważną jak uwierzytelnianie.	PEM zawsze projektowano specyficznie do pracy z pocztą elektroniczną. Uwierzytelnianie jest ważniejsze od prywatności (nie jest możliwe wysłanie wiadomości PEM, która nie jest uwierzytelniona).
Szyfrowanie	Obsługuje wiadomości szyfrowane i podpisane; może pod-pisać wiadomość nie szyfrowaną; może szyfrować wiadomości nie podpisane. Używa algorytmów IDEA, RSA, MD5.	Obsługuje wiadomości szyfrowane i podpisane; może podpisać wiadomość nie szyfrowaną; nie jest możliwe wysłanie wiadomości, która jest nie podpisana. Używa algorytmów DES, RSA, MD5.
Wytwarzanie kluczy	Użytkownik PGP wytwarza swoją własną parę klucz-jawny / klucz-prywatny, korzystając z posiadanego oprogramowania.	Certyfikaty według standardu X.509 są wytwarzane przez użytkownika, albo przez wyznaczoną osobę, albo przez urządzenie sprzętowe.
Ukrywanie informacji o podpisie	Niemożliwe jest sprawdzenie podpisu PGP, jeżeli wiadomość jest nadal zaszyfrowana.	Możliwe jest sprawdzenie podpisu PEM, nawet jeżeli wiadomość jest szyfrowana.
Dystrybucja kluczy	Klucze jawne PGP typowo rozprowadza się pocztą elektroniczną za pomocą sieciowych biuletynów informacyjnych lub serwerów kluczy, opartych na poczcie elektronicznej. W celu przeciwdziałania sabotażom klucze są podpisywane przez trzecią stronę - innych użytkowników.	"Urząd poświadczający" (ang. Certification Authority) jest obdarzoną powszechnym zaufaniem bazą kluczy jawnych. Klucze jawne PEM mogą być rozprowadzane za pomocą dowolnych środków, zwykle pocztą elektroniczną lub za pomocą przeglądania skorowidzów zbudowanych na podstawie standardu X.500.
Unieważnianie kluczy	Dystrybucja kluczy jest doraźna i w znacznej części rozpowszechniana ustnie. W tej sytuacji niemożliwe jest zagwarantowanie unieważnienia certyfikatu, jeżeli jest on skompromitowany.	Listy unieważnień certyfikatów (CRL) są zachowane w skorowidzach typu X.500 lub w skrzynkach pocztowych, utrzymywanych przez każdy urząd typu PCA.

Porównanie PGP i PEM

syłanych wiadomości. Ze względu na różnice związane zasadniczo z obsługą certyfikatów kluczy jawnych PGP nadaje się bardziej do bezpośredniej komunikacji między ludźmi na bazie wzajemnej znajomości i zaufania. Natomiast PEM ze względu na ścisłą hierarchię zarządzania kluczami jest przydatny dla instytucji i przedsiębiorstw. Jednak z powodu braku odpowiedniej infrastruktury do zarządzania kluczami celowe wydaje się podjęcie prac w zakresie oddolnej budowy, np. akademickich hierarchii poświadczania kluczy oraz aplikacji umożliwiających przenoszenie wiadomości między PGP i PEM.

Literatura:

[1] SCHNEIER B., ***Kryptografia dla praktyków***, WNT 1995.
[2] SCHNEIER B., ***Ochrona poczty elektronicznej***, WNT 1996.
[3] ***Zalecenia RFC 1421, 1422, 1423, 1424***, Internet Architecture Board, IETF PSRG, 1993.
[4] ZIMMERMAN P., ***PGP User Manual***, t. I i II, 1994.

Artykuł był prezentowany na II Konferencji „Informatyka na Wyższych Uczelniach dla Gospodarki Narodowej".

Zależności temporalne w modelowaniu i analizie procesów gospodarczych

Krzysztof Kania, Stanisław Kędzierski, Jerzy Gołuchowski

Klasyczne metodologie tworzenia systemów informatycznych zarządzania koncentrują się głównie na opisie statyki modelowanej dziedziny. Stosowane metody modelowania nie ujmują jawnie zmienności modelowanej dziedziny. Zmienność tę można i należy uwzględniać w modelu konceptualnym w taki sposób, aby była możliwa zarówno jawna specyfikacja zależności temporalnych w procesach gospodarczych, jak i ujęcie historyczne realizowanych procesów.

Systemy informatyczne zarządzania (SIZ) coraz częściej muszą być traktowane jako systemy czasu rzeczywistego. W warunkach rosnącej konkurencji znacznie większa uwaga kierownictwa przedsiębiorstwa skupiona jest na procesach, które muszą zostać ukończone w zadanym czasie. Dotyczy to zarówno prostych działań, takich jak sprawna obsługa klienta przy kasie, jak i złożonych, takich jak prowadzenie kompleksowych działań inwestycyjnych. Zadaniem SIZ jest więc coraz częściej nie tylko ewidencja i kontrola zasobów firmy, lecz także (a może przede wszystkim) wspomaganie realizacji procesów składających się na działalność przedsiębiorstwa.

Sprawne rozwiązywanie zadań bieżącej działalności przedsiębiorstwa za pomocą SIZ obejmuje tylko część prac związanych z zarządzaniem firmą. Kierownictwo przedsiębiorstwa bowiem musi analizować, prognozować, planować itp. działalność także w dłuższej perspektywie. Aby sprostać tym wymaganiom strategicznego myślenia, kierownictwo potrzebuje wiedzy wykraczającej poza podręcznikową wiedzę o przedsiębiorstwie. Źródłem jej może być analiza rozwoju zjawisk gospodarczych przebiegających w zarządzanym przedsiębiorstwie. SIZ powinny więc dostarczać dane oraz narzędzia zaspokajające te potrzeby. Bazy danych SIZ muszą więc opierać się na takim modelu świata, który koncentruje uwagę (podobnie jak ekonomista) na rozwoju (zmienności) rzeczywistości gospodarczej [2].

Analizując zatem potrzeby modelowania rzeczywistości w zależności od potrzeb budowy SIZ, dochodzimy do wniosku, że zależności temporalne występujące w modelowanym świecie można i należy uwzględniać w modelach konceptualnych w dwóch perspektywach czasowych, pozwalających na:

- jawną specyfikację zależności temporalnych zachodzących pomiędzy obiektami współdziałającymi w procesach gospodarczych;
- ujęcie historyczne (ewolucyjne) elementów rzeczywistości (bytów materialnych i procesów).

Wyróżnienie dwóch perspektyw czasowych w opisie rzeczywistości w związku z potrzebami budowy SIZ służy różnym celom:

- opisowi dynamiki - pozwala przedstawiać zależności temporalne nakładane na czynności (działania) oraz obiekty uczestniczące w procesie. Celem takiego ujęcia jest optymalizacja wykonania działań, rozwiązanie problemów wspólnego dostępu do zasobów itp.;
- prawidłowemu ujęciu w modelu konceptualnym zmian rozpatrywanych w dłuższej perspektywie (w aspekcie historycznym) - pozwala na właściwe gromadzenie i udostępnianie przetworzonej wiedzy o rozwoju zjawisk (procesów lub obiektu) użytkownikom badającym rzeczywistość. Takie badania przeprowadza się w toku diagnozy, wykrywania praw rozwoju, reorganizacji procesów, restrukturyzacji przedsiębiorstw itp.

Aby budować SIZ, które są w stanie podołać tak postawionym zadaniom, konieczne jest rozszerzenie metodologii tworzenia SIZ w zakresie zakładanej ontologii przedsiębiorstwa oraz formalizmów modelowania procesów i historii. Dodatkowymi pojęciami ontologicznymi, które - jak się wydaje - należy uwzględnić w metodologii tworzenia SIZ, są pojęcia: historii (rozwoju) i dynamiki (współdziałania).

Modele konceptualne temporalnych baz danych

W modelach konceptualnych klasycznych BD nacisk kładzie się przede wszystkim na ujęcie statycznych elementów (encji, związków) budowanych systemów [4]. Znaleźć w nich zatem można odpowiedź na pytanie, co obiekt ujmuje. Tymczasem rzeczywistość jest zmienna i bardzo często jej analiza wymaga uwzględnienia zachodzących w niej zmian. Należy więc byty odwzorowywane w modelu konceptualnym traktować jako twory zarówno działające (współdziałające), jak i rozwijające się w czasie (posiadające swoją historię). Zachodzi więc potrzeba odpowiedzi - nie tylko w toku analizy rzeczywistości,

ale już w trakcie tworzenia modelu konceptualnego - na pytania:

- co może się zmienić w modelowanej rzeczywistości?
- jak się może zmienić?
- jak zapisać rodzaj tych zmian w modelu konceptualnym, a następnie w bazach danych SIZ?

Przez dynamikę (systemu) będziemy rozumieć taki opis rzeczywistości, który uwzględnia to, że poszczególne procesy zachodzące w systemie mogą być rozpatrywane w temporalnych zależnościach pomiędzy elementami wchodzącymi w jego skład i ograniczonych długością trwania procesu. Zjawiska analizowane są więc w relatywnie krótkich odcinkach czasu. Przykładowo, w aspekcie dynamiki można rozpatrywać na przykład proces tworzenia dokumentu biurowego w tym sensie, że poszczególne składowe dokumentu muszą być wytwarzane w określonej kolejności. Ich wytworzenie zajmuje ustalony czas. Pewne elementy mogą być tworzone równolegle, istnieje ścieżka krytyczna realizowanego procesu itd.

Przez historię (systemu) będziemy rozumieć opis ewolucji obiektu występującego w realnym świecie i odwzorowanego w bazie danych. Obiektem tym może być rzecz, osoba, proces itp. W modelu dziedziny przedmiotowej, obejmującej przedsiębiorstwo w zależności od rodzaju analizowanego obiektu, wyróżnić można różne rodzaje historii: historie wartości atrybutu, historię wystąpienia, historie przebiegów procesów, historię relacji, historię schematu bazy danych, historie transakcji itd. Zjawiska rozpatrywane są więc w długiej perspektywie czasu. W perspektywie historycznej dostrzega się ewolucję procesów tworzenia dokumentów: jak może się zmieniać rola dokumentu w realizowanych procesach, jakie są tego konsekwencje, jak zmieniają się powiązania z innymi dokumentami itp. W ujęciu historycznym można więc rozpatrywać (powtarzalne) procesy jako obiekty, które posiadają swoją historię (jeden trwał dłużej, inny krócej, w realizacji jednego brał udział pracownik X, w innym nie itd.).

W systemach ekonomicznych często można obserwować zmiany, dotyczące m.in.:

- składu procesów;
- logiki procesów (struktury realizowanych procesów);
- warunków (zewnętrznych) ich realizacji;
- agentów realizujących poszczególne działania w procesie.

W toku zarządzania nie wystarcza, aby wymienione rodzaje zmian traktować jedynie w kategoriach statystycznych. Chcąc zdobyć wiedzę głęboką (np. o związkach przyczynowych), konieczna jest szczegółowa analiza danych historycznych o realizacji procesów. Powstaje więc problem opisania w modelu konceptualnym związków zachodzących pomiędzy procesami oraz procesami a uczestniczącymi w nich obiektami również w ujęciu historycznym [5]. Taki opis powinien stanowić ogniwo łączące użytkowników z konstruktorami baz danych.

Atrybuty temporalne w bazach danych

Z punktu widzenia potrzeb odwzorowania zmienności świata wyróżniane obecnie w SZBD typy atrybutów (danych) nie są wystarczające. Zachodzi potrzeba, aby atrybuty były postrzegane także ze względu na ich zmiany w czasie. Trzeba zatem atrybuty podzielić na dwie grupy:

- niezmienne - takie, które nie zmieniają się w czasie;
- zmienne (temporalne) - takie, których zmiany wartości w czasie powinny być odnotowane w systemie.

Obecnie ***atrybuty temporalne,*** ze względu na brak wyodrębnionego ich typu, tworzy się jako atrybuty złożone. Zatem obecnie zarówno atrybuty niezmienne jak i zmienne w czasie nie są od siebie odróżniane i są tworzone za pomocą tych samych typów danych, z tą różnicą, że atrybuty temporalne są powiązane (lecz tylko umownie i nieformalnie) z odpowiednimi zapisami określeń czasu.

Reprezentacja atrybutów temporalnych w klasycznych bazach ogranicza się do ich złożenia z atrybutów prostych, których znaczenie definiowane jest w relacji przez projektantów i programistów SI (użytkowników SZBD). Przypisują oni poszczególnym atrybutom określone znaczenia (data urodzenia, data zatrudnienia, czas zmiany ceny, czas zmiany stanowiska itd.). Sami też konstruują odpowiednie procedury obsługi tego rodzaju informacji. Taki sposób posługiwania się informacją temporalną określa się jako indywidualny sposób użytkownika (*user-defined time*) (np. [3]). Tak skonstruowany model zmiennej (ewoluującej) rzeczywistości nie jest niestety naturalny, a zrozumienie go przez użytkownika nie mającego doświadczenia w programowaniu - wątpliwe. Zawiłe konstrukcje oraz dowolność w interpretacji niektórych opisów spowodować mogą zerwanie więzi pomiędzy widzeniem świata przez użytkownika a jego obrazem zawartym w bazie danych.

Częściowym rozwiązaniem problemów związanych z reprezentacją atrybutów temporalnych w tradycyjnych systemach baz danych jest wydzielenie osobnego typu danych służących tylko do rejestracji zmian innych wartości w bazie. Atrybuty rejestrujące zmienność dziedziny przedmiotowej powinny zostać oddzielone od atrybutów służących do przechowywania wartości statycznych.

Dalej idącą propozycją przezwyciężenia ograniczeń tradycyjnego modelu danych jest wprowadzenie zbioru nowych, bardziej złożonych typów danych. Przedstawia je tabela 1 (por. także [4], [11]).

Rodzaj wartości	Opis	Przykład
identyfikatory	niezmienne w czasie identyfikujące obiekty	nr PESEL
niezmienne w czasie	stałe, niezależne od czasu	miejsce i data urodzenia, imiona rodziców
zmienne w czasie	dane, które mogą dezaktualizować się	adres zamieszkania, stanowisko
określające czas	dane określające czas	data urodzenia, data zatrudnienia
nakładające się	dane opisujące kolejno nabywane własności obiektu	znajomość języków, kursy zawodowe
czasowo określone	dane opisujące dane w przyszłości	tymczasowe uprawnienia, próbny okres zatrudnienia

Tabela 1. Typy danych w odniesieniu do czasu

Wprowadzenie wyodrębnionych atrybutów temporalnych na poziom modelowania konceptualnego i logicznego pozwoli użytkownikom na jednoznaczne opisanie zmienności zachodzącej w dziedzinie przedmiotowej i formalne wyrażenie charakteru tych zmian.

Jednym ze sposobów przezwyciężenia trudności ujęcia zmienności modelowanej dziedziny w bazie danych jest konsekwentne ujmowanie jej (przez odpowiednie ujęcie czasu) na wszystkich etapach (poziomach) tworzenia SI. Narzędzi do rozwiązania tego problemu dostarcza technologia temporalnych baz danych (zob. szerzej [6]). Na poziomie konceptualnym umożliwia ona (por. np. [8],[9]):

- wprowadzenie do modelu odpowiedniej semantyki i ontologii czasu;
- wskazanie powiązań pomiędzy obiektami (encjami oraz związkami) a czasem.

Na poziomie logicznym pozwala ona na:

- skonstruowanie odpowiednich struktur danych służących wygodnej dla użytkowników reprezentacji atrybutów temporalnych;
- wprowadzenie działań na tych strukturach umożliwiających swobodne manipulowanie informacją o charakterze temporalnym.

Potrzeba logicznej reprezentacji zależności temporalnych w opisie procesów

Systemy wspomagające procesy zachodzące w praktyce gospodarczej muszą uwzględniać zależności i ograniczenia czasowe, które mogą wystąpić w rzeczywistości. Ograniczenia te są znacznie łagodniejsze niż w systemach technicznych sterowania i regulacji. Wynika to z innego rodzaju praw rządzących przebiegiem tych procesów. W świecie procesów gospodarczych zachowania poszczególnych podmiotów podlegają również prawom ustalonym przez człowieka, a nie tak,jak w systemach technicznych(tylko prawom natury).

Prawa ustanowione przez ludzi można podzielić na zewnętrzne względem danej instytucji (np. konstytucja, ustawy, rozporządzenia) i wewnętrzne (np. zarządzenia, instrukcje, regulaminy), które stanowią podstawę (bazę normatywną) do konstruowania procesów zachodzących w przedsiębiorstwie (agentów, ich uprawnień, procedur, czasu realizacji, zasobów). Trzeba przyjąć, że w danym momencie obowiązuje wspólne i niesprzeczne wewnętrznie prawo. Jest to jedno z założeń umożliwiających modelowanie świata idealnego. Takie ujęcie przedsiębiorstwa (jako zbioru agentów, uprawnień, ról, koniecznych do wykonania czynności) wymaga narzędzi umożliwiających formalne przedstawienie w modelu:

- prawnych norm postępowania;
- obowiązków i zakazów wynikających z zaistnienia określonej sytuacji;
- agentów realizujących proces;
- czynnika czasu.

Wymienione wymagania skłoniły do stworzenia temporalnej relatywnej logiki deontycznej do modelowania procesów zachodzących w przedsiębiorstwie [7].

Logika deontyczna jako narzędzie modelowania zachowań normatywnych

Logika obowiązku, przyzwolenia i zabronienia nazywana jest logiką deontyczną, logiką norm lub logiką systemów normatywnych [10], [14]. Aksjomaty i reguły logiki deontycznej według von Wrighta [12] zawierają:

(OS0) wszystkie (wystarczające) tautologie rachunku zdań,

(OS1) $O\,p \equiv \neg P \neg p$

(OS2) $P\,p \vee P \neg p$

(OS3) $P\,(p \vee q) \equiv P\,p \vee P\,q$

(OS4) $p \equiv q \Rightarrow P\,p \equiv P\,q$

Aksjomat (OS1) wyraża fakt, że obowiązek p jest równoważny nie pozwalaniu na nie p; (OS2) mówi, że dozwolone jest p lub nie p; (OS3) mówi, że dozwolenie p lub q jest równoważne dozwoleniu p lub dozwoleniu q; (OS4) mówi, że jeśli dwa wyrażenia są równoważne, wtedy dozwolenie jednej implikuje dozwolenie drugiej i odwrotnie.

Relatywna logika deontyczna

Rozszerzeniem klasycznego rachunku deontycznego jest relatywny rachunek deontyczny. Zezwala on na konstruowanie wypowiedzi o obowiązku warunkowym. W klasycznym rachunku deontycznym nie jest możliwe sformalizowanie wypowiedzi o obowiązku warunkowym za pomocą formuł typu: $O(A \rightarrow B)$ lub $O(p \rightarrow q)$. Z tego względu zamiast przedstawionych formuł wprowadza się nową formułę zastępującą formułę OX (gdzie X oznacza zmienną zastępującą nazwę czynności lub stanu rzeczy). Formuła ta ma postać: $O(X / Y)$, gdzie X i Y oznaczają czynności i stany rzeczy lub są wyrażeniami będącymi połączeniem spójnikami zmiennych oznaczających czynności i stany. Spójniki te zapisywane są identycznie jak spójniki rachunku zdań, a zmienne łączone są nimi analogicznie jak zmienne rachunku $O(X / Y)$.

Relatywny rachunek deontyczny został wprowadzony w roku 1964 przez von Wrighta [13]. W rachunku tym formułę $O(A / B)$ czyta się następująco: należy sprawić to, że A, gdy B lub jako: powinno się sprawić stan rzeczy A, gdy B, względnie - gdyby zachodził stan rzeczy B. Aksjomatami przyjętymi, są:

A1 $\neg[O(A / B) \wedge O(\neg A / B)]$

A2 $O(A \wedge B / C) \leftrightarrow O(A / C) \wedge O(B / C)$

A3 $O(A / B \vee C) \leftrightarrow O(A / B) \wedge O(A / C)$

A4 $P(A / B) \leftrightarrow \neg\, O(\neg A / B)$

W codziennej praktyce pracy przedsiębiorstwa występuje jednak wiele sytuacji, które zależą od zmieniających się regulacji prawnych lub od upływającego czasu. Przykładowo ta sama sprawa przedłożona w urzędzie parę dni później może być za-

kończona w inny sposób, nie naruszając praw obowiązujących w rozpatrywanym czasie. Jeśli ktoś przedłoży swoje podanie przed dniem wejścia w życie określonych przepisów prawnych, sprawa może zostać odrzucona. Przedłożenie sprawy po dniu wejścia w życie nowych regulacji prawnych spowoduje pozytywne rozpatrzenie sprawy. Pewne operacje bowiem mogą być niedopuszczalne przed określonym terminem lub po nim.

Przedstawiony przykład wskazuje, że istnieje potrzeba modelowania procesów, w których opisie normatywnym muszą występować zależności czasowe. Z tego też powodu wprowadzamy nowy rodzaj relatywnej logiki deontycznej - temporalną relatywną logikę deontyczną (TRLD). Opiera się ona na aksjomatyzacji czasu, przedstawionej przez Allena [1].

Temporalna relatywna logika deontyczna

Formalne włączenie czasu w predykaty relatywnej logiki deontycznej rozszerza ją do temporalnej relatywnej logiki deontycznej. Uprzednio zdefiniowane predykaty O, F i P przyjmują więc następującą postać:

O_T (X/Y) oznacza: obowiązkowe jest X pod warunkiem spełnienia Y w czasie T,

F_T (X/Y) oznacza: zabronione jest X przy spełnionym warunku Y w czasie T,

P_T (X/Y) oznacza: dopuszczalne jest X przy spełnionym warunku Y w czasie T.

Subskrypt T może oznaczać - w zależności od interpretacji - nie tylko punkt czasu T, ale także, a może nawet przede wszystkim, przedział czasu ($t_1 < t < t_2$) albo też czas przed chwilą t_0 ($t < t_0$) lub po chwili t_0 ($t > t_0$).

Kolejnym rozszerzeniem, które wprowadzamy do relatywnej logiki deontycznej jest wskazanie agenta (lub grupy agentów), którego dotyczy operator deontyczny. Ponieważ czynności X i Y mogą być wykonywane przez różnych agentów, korzystne jest jednoznaczne wskazanie agentów odpowiedzialnych za wykonanie X i Y. Odpowiedni zapis operatorów deontycznych przybiera następującą postać:

O_T (X(p)/Y(r)) oznacza: obowiązkowe jest wykonanie X przez agenta p w czasie T, pod warunkiem uprzedniego spełnienia Y przez agenta r;

F_T (X(p)/Y(r)) oznacza: zabronione jest wykonanie X przez agenta p w czasie T, przy uprzednio spełnionym warunku Y przez agenta r;

P_T (X(p)/Y(r)) oznacza: dopuszczalne jest wykonanie X przez agenta p w czasie T, przy spełnionym warunku Y przez agenta r.

Oczywiście może się zdarzyć, że warunek Y nie ma jawnie podanego (wskazanego) wykonawcy. Przykładowo warunki, takie jak: upływ czasu lub osiągnięcie odpowiednich warunków klimatycznych należą do tego typu ograniczeń wstępnych. W tym wypadku dopuszczalne jest opuszczenie symbolu agenta odpowiedzialnego za warunek.

W modelowaniu przedsiębiorstwa normy odpowiadają przepisom i ogólnie przyjętym zwyczajom. Opis ciągów działań składających się na rozpatrzenie konkretnej sprawy można uzyskać za pomocą zdań zapisanych w TRDL.

❖ ❖ ❖

W artykule pokazano, że klasyczne metodologie tworzenia systemów informatycznych zarządzania, koncentrujących się na opisie statyki modelowanej dziedziny, są niewystarczające do modelowania rzeczywistości. Zaproponowano wprowadzenie dwu perspektyw czasowych, określonych w pracy jako dynamika systemu oraz historia systemu. Przedstawiono propozycje rozwiązań trudności odwzorowania zmienności w konceptualnych modelach baz danych temporalnych oraz deklaratywnego ujęcia procesu za pomocą logiki norm.

Wprowadzona temporalna relatywna logika deontyczna wydaje się stanowić dobre narzędzie do modelowania procesów gospodarczych. Pozwala ująć w opisie procesu zarówno zależności modalne, jak i temporalne, a także osobę (agenta) odpowiedzialną za wykonanie określonych czynności.

Literatura:

[1] Allen J., ***Maintaining knowledge about temporal intervals***. CACM, vol.26, no. 11, 1983.

[2] Gołuchowski J., Kajstura T., Kania K., Kędzierski S., ***Modelowanie procesów gospodarczych jako podstawa zintegrowanych systemów informatycznych zarządzania***. INFOGRYF'96 (w druku).

[3] Jensen C.S. et al., ***A Consensus Glossary of Temporal Database Concepts***. SIGMOD RECORD, vol. 23, no. 1, March 1994.

[4] Kania K., Gołuchowski J., ***Zagadnienia reprezentacji czasu w klasycznych systemach baz danych***. Informatyka, nr 7, 1996.

[5] Kania K., Gołuchowski J., ***Reprezentacja czasu w modelu konceptualnym SI***. „Informatyka", nr 8, 1996.

[6] Kania K., Gołuchowski J., ***Systemy temporalnych baz danych***. „Informatyka", nr 9, 1996.

[7] Kędzierski S., ***Office modelling through deontic logic***. in S. Wrycza, J. Zupančič /eds/, ***Proceedings of the V Int. Conf. Information Systems Development*** - ISD'96. Methods & Tools. Theory & Practice, Gdańsk, Poland, 24 - 26 September 1996.

[8] Ling D., Bell D., ***Taxonomy of Time Models in Databases. Information and Software Technology***. vol. 32, no. 3, 1990.

[9] Snodgrass R., Ahn I., ***Temporal Databases***. Computer vol. 19, September 1986.

[10] Świrydowicz K., ***Logiczne teorie obowiązku***. Wydawnictwo Naukowe UAM, Poznań, 1995.

[11] Tansel A., ***Modelling temporal data. Information and Software Technology***, vol. 32, no. 8, 1990.

[12] von Wright G., ***Deontic logic***. Mind, vol. 60, 1951.

[13] von Wright G., ***A New System of Deontic logic***. Danish Yearbook of Philosophy. vol. I, 1964.

[14] Ziemba Z., ***Analityczna teoria obowiązku. Studium z logiki deontycznej***, Warszawa, 1983.

Artykuł był prezentowany na II Konferencji „Informatyka na Wyższych Uczelniach dla Gospodarki Narodowej".

POLSKIE TOWARZYSTWO INFORMATYCZNE

BIULETYN

NUMER 3 (152) ROK XVII MARZEC 1998

Pragnę poinformować Szanownych Czytelników, że do Redakcji Biuletynu PTI wpłynęło przemiłe zaproszenie od przedstawiciela Komisji Konkursowej, prof. Zygmunta Mazura, na uroczystość ogłoszenia wyników i rozdania nagród laureatom Ogólnopolskiego Konkursu na najlepsze prace magisterskie z informatyki. Uroczystość odbyła się 12 lutego 1998 roku o godz. 12.00 w sali seminaryjnej Wydziałowego Zakładu Informatyki Politechniki Wrocławskiej. To już XIV edycja tego Konkursu, który promuje wyróżniające się prace młodych absolwentów kierunku informatyka. Redakcja serdecznie gratuluje laureatom oraz ich promotorom.

Posypały się też gratulacje na liście dyskusyjnej PTI-L. Koledzy z Towarzystwa z dużym uznaniem przyjęli fakt ufundowania wszystkich tegorocznych nagród przez firmę INTERNATIONAL BUSINESS SYSTEMS z Wrocławia. Stało się to za sprawą prezesa firmy, pana dr. Andrzeja Krupy.

Ewa Łukasik

Wyniki XIV Ogólnopolskiego Konkursu na najlepsze prace magisterskie z informatyki, zorganizowanego w 1997 roku przez Polskie Towarzystwo Informatyczne

Rozstrzygnięto XIV Ogólnopolski Konkurs PTI na najlepsze prace magisterskie z informatyki. Do konkursu zgłoszono 24 prace, wykonane w roku akademickim 1996/97 w dziewięciu krajowych wyższych uczelniach: w Akademii Ekonomicznej we Wrocławiu (1), w Akademii Medycznej w Łodzi (1), w Politechnice Gdańskiej (1), w Politechnice Poznańskiej (5), w Politechnice Śląskiej w Gliwicach (3), w Politechnice Warszawskiej (2), w Politechnice Wrocławskiej (5), na Uniwersytecie Jagiellońskim (3) oraz na Uniwersytecie Wrocławskim (3).

Na posiedzeniu w dniu 22 grudnia 1997 roku we Wrocławiu Komisja Konkursowa w składzie: prof.dr hab.inż. Czesław Daniłowicz (przewodniczący), prof.dr hab.inż. Zbigniew Huzar, prof.dr hab.inż. Jan Magott, dr hab. Andrzej Marciniak, prof. dr hab. Zygmunt Mazur, dr inż. Zbigniew Szpunar (sekretarz), uwzględniając opinie recenzentów prac konkursowych, po dyskusji ustaliła następujące rozstrzygnięcie konkursu:

Pierwszą nagrodę, w wysokości 1,200 zł, otrzymał

mgr inż. Krzysztof GIARO

za pracę pt. *Badanie zwartego szeregowania zadań na procesorach dedykowanych,*

wykonaną w Politechnice Gdańskiej (Wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Podstaw Informatyki; promotor: prof.dr hab. Marek Kubale).

Drugą nagrodę, w wysokości 900 zł, otrzymał

mgr inż. Marcin GRABOŃ

za pracę pt. *Przestrzenna wizualizacja ukształtowania terenu,*

wykonaną w Politechnice Wrocławskiej (Wydział Informatyki i Zarządzania, Wydziałowy Zakład Informatyki; promotor: dr inż. Jerzy Sas).

Trzecią nagrodę, w wysokości 700 zł, otrzymał

mgr inż. Daniel SŁOBODZIAN

za pracę pt. *Zarządzanie transakcjami w architekturach klient-serwer obiektowych baz danych,*

wykonaną w Politechnice Poznańskiej (Wydział Elektryczny, Instytut Informatyki; promotor: prof.dr hab.inż. Tadeusz Morzy).

Trzy równorzędne wyróżnienia po 500 zł otrzymali:

mgr inż. Adam CZAJKA

za pracę pt. *Projektowanie oprogramowania metodą HRT-HOOD i jego realizacja w języku C,*

wykonaną w Politechnice Poznańskiej (Wydział Elektryczny, Instytut Informatyki; promotor: dr hab.inż. Jerzy R. Nawrocki).

mgr Michał KĘPIŃSKI

za pracę pt. *Przetwarznie mowy w interface'ie użytkownika dla osób niepełnosprawnych,*

wykonaną w Uniwersytecie Jagiellońskim (Wydział Matematyki i Fizyki, Instytut Informatyki; promotor: dr inż. Andrzej Izworski z AGH w Krakowie),

mgr inż. Krzysztof OBORZYŃSKI

i mgr inż. Cezary SOBANIEC

za pracę pt. *Projekt i realizacja systemu z rozproszoną pamięcią współdzieloną,*

wykonaną w Politechnice Poznańskiej (Wydział Elektryczny, Instytut Informatyki; promotor: prof.dr hab.inż. Jerzy Brzeziński).

Dla autorów trzech nagrodzonych prac Polskie Towarzystwo Informatyczne ufundowało ponadto udział w Szkole PTI w Szczyrku (czerwiec 1998 r.), gdzie laureaci będą mogli zaprezentować swoje prace.

Laureatom konkursu składamy gratulacje oraz życzenia sukcesów w dalszej pracy zawodowej.

W imieniu organizatorów XIV konkursu składamy podziękowanie Panu dr. Andrzejowi KRUPIE, prezesowi firmy INTERNATIONAL BUSINESS SYSTEMS z Wrocławia za ufundowanie wszystkich nagród na łączną kwotę 4300 zł.

Sekretarz Komisji
dr inż. Zbigniew Szpunar

Europejskie Komputerowe Prawo Jazdy

Akcja ECDL w Polsce systematycznie rozwija się. Od czasu do czasu drukujemy informacje o kolejnych egzaminach. Dziś Kol. Marka Miłosz, koordynator akcji ECDL w Polsce, przesyła „na żywo" wiadomość z pola walki o ECDL.

W Liceum Ogólnokształcącym w Piasecznie Kol. Basia Szymańska aktywnie egzaminuje na ECDL.

O rezultatach jej pracy i nauczycielki informatyki w tej szkole niech świadczy poniższy list:

„Potwierdzam wiadomość, że dziewięciu uczniów zakończyło egzaminy, zaliczając po 7 modułów.

Oto nazwiska uczniów:

1. Dariusz Dobrowolski
2. Marcin Ciura
3. Piotr Markowski
4. Marcin Dąbrowski
5. Jakub Zaremba
6. Przemysław Wasiewicz
7. Jakub Widłak
8. Arkadiusz Guzicki
9. Małgorzata Wasiewicz

.....

Po feriach (semestr zaczynamy 23 lutego) chciałabym oficjalnie wręczyć Certyfikaty uprawnionym uczniom.

Małgorzata Białas"

Mamy już za sobą prawie 1000 egzaminów i ponad 300 osob uczestniczących w akcji ECDL.

Proszę wszystkich członków i sympatyków PTI o dalsze aktywne działania.

Marek Miłosz - koordynator akcji ECDL w Polsce

II Regionalna Konferencja Informatyczna w Kazimierzu nad Wisłą

Koło PTI w Lublinie współorganizuje w końcu maja II Regionalną Konferencję Informatyczną w Kazimierzu nad Wisłą. Konferencja jest pomyślana jako cykl różnych imprez powiązanych ze sobą głównie przez PTI i tematykę informatyczną.

W dniach 25-29.05.98 odbędą się:

- II Lubelskie Akademickie Forum Informatyczne;
- VII Konferencja Polskiego Klubu Użytkowników Systemu PROGRESS;
- I Konferencja „Informatyk Zakładowy";
- VII Warsztaty Mikrokomputerowe Politechniki Lubelskiej;
- Konferencje metodyczne nauczycieli Elementów Informatyki.

Serwis Konferencji (w budowie) dostepny jest pod adresem:
http://pluton.pol.lublin.pl/kaz98.htm

Wszystkich zainteresowanych
SERDECZNIE ZAPRASZAMY

Marek Miłosz

Reminiscencje konferencji „Informatyka w administracji samorządowej"

W dniach 15 - 17 października 1997 w Ustroniu koło Bielska -Białej odbyła się druga konferencja „Informatyka w administracji samorządowej ", współorganizowana przez Oddział Górnośląski PTI oraz katowickie biuro Fundacji Rozwoju Demokracji Lokalnej.

Impreza miała charakter spotkania osób odpowiedzialnych za zakup i eksploatację systemów informatycznych w urzędach gminnych i miejskich z firmami działającymi na tym obszarze oraz osobami zajmującymi się eksploatacją tego rodzaju oprogramowania. Ogółem wystąpiło dziewięć firm z terenu całego kraju. Ponadto cztery firmy uruchomiły stałe stanowiska prezentacyjne w celu demonstrowania swoich rozwiązań . W konferencji wzięło udział około 70 osób w charakterze słuchaczy oraz 30 pracowników firm i wykładowców . Biorąc pod uwagę niewielki nakład środków na promocję konferencji i pokaźny wzrost liczby uczestników w stosunku do roku 1996 (ok. 30%), można mówić o sukcesie .

Głównym tematem wykładów była problematyka systemów obiegu dokumentów. Tej sprawie poświęciły swoje wykłady firmy Rodan - System Warszawa, Logotech Mysłowice i COIG Katowice. Wystąpili także wykładowcy niezależni, którzy omówili systemy obiegu dokumentów z punktu widzenia ich organizacji i projektowania.

Przewidujemy dalszy wzrost zainteresowania systemami informatycznymi przygotowywanymi dla obsługi administracji , w tym także administracji samorządowej. Aktualnie przygotowywana reforma administracji samorządowej oraz nowe akty prawne w zakresie ochrony danych osobowych, dokumentu i podpisu elektronicznego itd. powodują , że rynek w tym zakresie zmienia się i poszerza. Rosnąca zamożność urzędów administracji i coraz większe zrozumienie potrzeby inwestowania w informatykę sprawiają, że stale nowe firmy poszukują klientów właśnie w administracji samorządowej. Dlatego przewidujemy, że trzecia konferencja na temat informatyki w administracji samorządowej, która odbędzie się w dniach 14-16 października br. zostanie zorganizowana w nowym miejscu, dającym możliwość zwiększenia liczby uczestników. Będzie to, również w Ustroniu, dom wczasowy „Jaskółka".

Przewidujemy kontynuację dotychczasowej tematyki z oczywistym „hitem", jakim jest w tym roku reforma samorządowa.

Zadbaliśmy także, jako organizatorzy, o program rozrywkowo- towarzyski. Udział w tego rodzaju konferencjach pozwala na bliższy kontakt firm z potencjalnymi odbiorcami, a także na nawiązanie bezpośrednich kontaktów ludzi pracujących w podobnych warunkach i mających podobne zainteresowania. Sądzimy, że z tego właśnie względu formuła konferencji, w odróżnieniu do targów, ma swoje miejsce na rynku i jest atrakcyjna także na przyszłość.

Ewa Pytasz FRDL Katowice
Wojciech Głazek PTI Katowice

„Informatyka" w każdym domu (członka PTI)

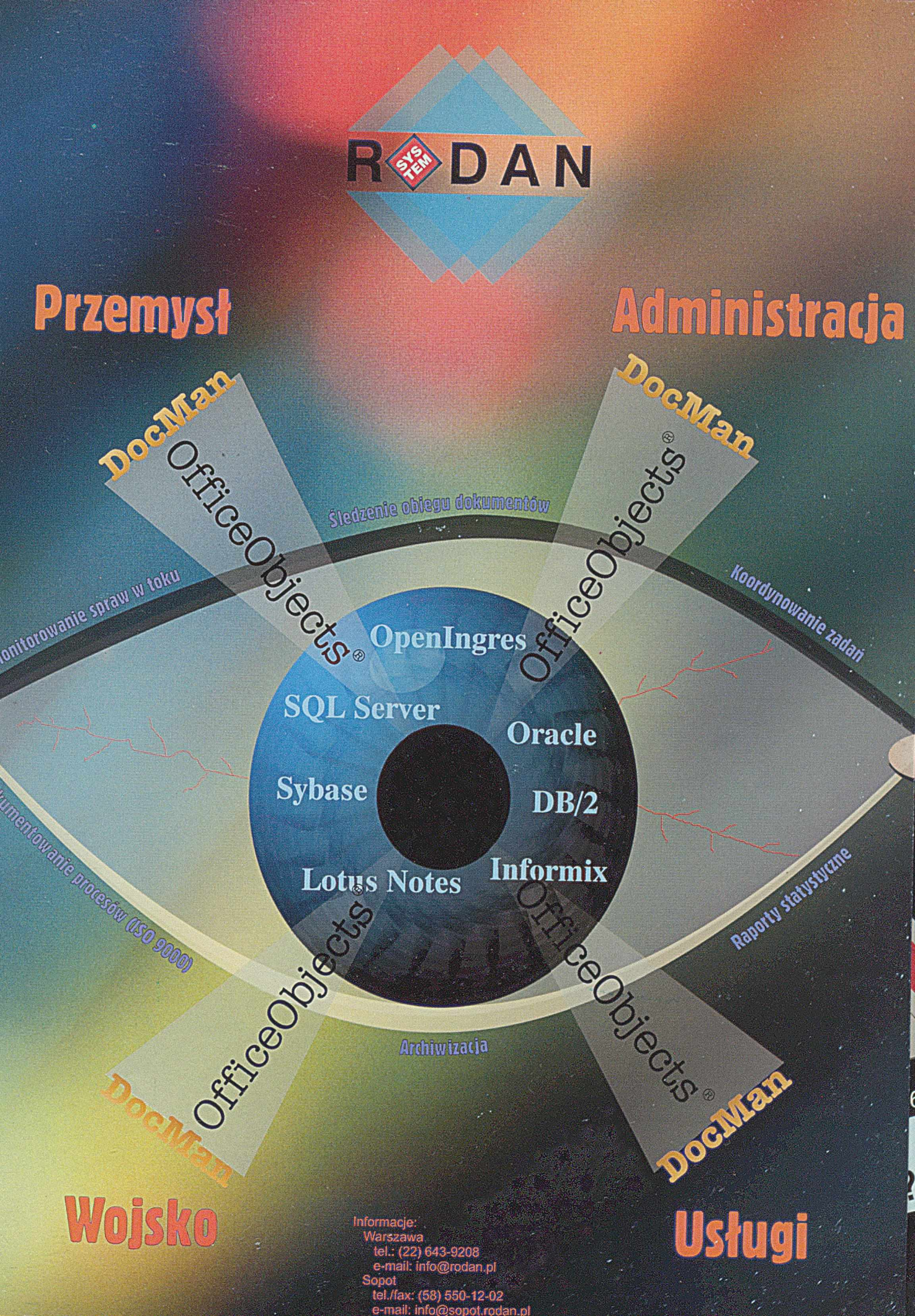
Zgodnie z zapowiedzią i decyzją Zarządu Głównego „Informatyka" wraz z Biuletynem PTI jest rozsyłana pilotażowo do domów wszystkich członków PTI. To dobrze czy źle? Redakcja ciekawa jest opinii Koleżanek i Kolegów. Jedna z pierwszych uwag dotarła do nas za pośrednictwem listy dyskusyjnej PTI-L. Pochodzi z Niemiec i brzmi następująco:

Przesyłanie miesięcznika **Informatyka** wszystkim członkom Towarzystwa jest wspaniałym pomysłem, ponieważ zawiera od pewnego czasu Biuletyn PTI. Dzięki temu członkowie, którzy nie mają na co dzień kontaktu z Kołami, dowiadują się, co się dzieje w Towarzystwie. Tak trzymać!

M. S.

Redaktor: EWA ŁUKASIK
e-mail: lukasik@put.poznan.pl, tel. (0-61) 878 23 73
Instytut Informatyki, Politechnika Poznańska,
ul. Piotrowo 3a, 60-965 Poznań

Polskie Towarzystwo Informatyczne, Zarząd Główny
tel. (22) 624 60 61 w.328, tel./fax (22) 652 32 59
URL: http://www.pol.pl/pti
ul. Żelazna 87, 00-879 Warszawa


RODAN
SYSTEM
Przemysł
Administracja
DocMan
OfficeObjects®
DocMan
OfficeObjects®
Śledzenie obiegu dokumentów
onitorowanie spraw w toku
Koordynowanie zadań
OpenIngres
SQL Server
Oracle
Sybase
DB/2
Informix
Lotus Notes
umentowanie procesów (ISO 9000)
Raporty statystyczne
OfficeObjects®
OfficeObjects®
Archiwizacja
DocMan
DocMan
Wojsko
Usługi
Informacje:
Warszawa
tel.: (22) 643-9208
e-mail: info@rodan.pl
Sopot
tel./fax: (58) 550-12-02
e-mail: info@sopot.rodan.pl

Nie ma znaczenia czy jesteś duży czy mały.
Ważne jest, jak się rozwijasz.

Każda firma, duża czy mała chce osiągnąć sukces. Sukces i ciągły wzrost to nie tylko właściwy sposób myślenia i planowania w przedsiębiorstwie, zarządzania ludźmi i środkami materialnymi, lecz również umiejętność przystosowywania się do zachodzących w otoczeniu zmian.

Sukces to także gwarancja szybkiej wymiany informacji między wszystkimi pracownikami biorącymi udział w różnych etapach procesu technologicznego.

W jego osiąganiu pomagają narzędzia, takie jak system SAP R/3.

System R/3 jest zbiorem aplikacji integrujących procesy gospodarcze w całym przedsiębiorstwie, począwszy od zarządzania finansami, poprzez rozliczanie kosztów, planowanie i organizację produkcji, sprzedaży, gospodarkę materiałową, remonty, planowanie inwestycji do zarządzania projektami i kadrami firmy. System R/3 pozwala działać szybko i efektywnie. Jest dostępny w polskiej wersji językowej i jest dostosowany do polskich regulacji prawnych. Wdrażają go w Polsce wykwalifikowani konsultanci.

R/3 jest systemem standardowym, wykorzystującym najnowocześniejsze technologie informatyczne, niezależnym od platformy sprzętowej i systemowej oraz mającym stabilną, modułową konstrukcję. Daje to gwarancję niezawodności i bezpieczeństwa inwestycyjnego, a jednocześnie sprawia, że okres wdrożenia jest relatywnie krótki.

Z drugiej strony, dzięki szerokim możliwościom parametryzacji i adaptacji, można dostosować go do indywidualnej struktury organizacyjnej każdego przedsiębiorstwa i instytucji.

SAP R/3 pomaga prawidłowo zarządzać przedsiębiorstwem. Teraz i w przyszłości. Znajduje to potwierdzenie już w ponad 11000 instalacji systemu R/3 na całym świecie, niezależnie od wielkości i charakteru działalności firmy.

Jeżeli macie Państwo pytania dotyczące systemu R/3, chętnie odpowiemy na nie i prześlemy materiały informacyjne.

Nasz adres:
SAP Polska Sp. z o.o. Mokotów Business Park
02-672 Warszawa, ul. Domaniewska 41
tel. (+48 22) 606 06 06; fax (+48 22) 606 06 07

SAP®
A Better Return On Information.